*(PRELIMINARY)*

# DDP-124
## Programmers Reference Manual

COMPUTER CONTROL COMPANY, INC.

Document No. 71-398/P

# PRELIMINARY

# PROGRAMMERS REFERENCE MANUAL

# FOR THE

# DDP-124

# GENERAL PURPOSE COMPUTER

August 1965

Computer Control Company, Inc.
Old Connecticut Path
Framingham, Massachusetts

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)



LIST OF ILLUSTRATIONS

## LIST OF TABLES

DDP-124 GENERAL PURPOSE COMPUTER

DDP-124 General Purpose Computer

# SECTION I
# INTRODUCTION AND DESCRIPTION

## INTRODUCTION

This manual is one of a series of support documents for the DDP-124 General-Purpose Computer and is intended to provide the information that a programmer will normally require in preparing and running programs on the computer. Other documents in the DDP-124 series that will be of interest to programmers are as follows.

DAP II Manual for the DDP-124
DDP-124 Programming Bulletins
FORTRAN II Manual for the DDP-124

Programming the DDP-124 is very similar to programming most of the popular single-address binary computers. For this reason, it is felt that no major idiosyncrasies confront the programmer who is new to the DDP-124. Also, this manual presents programming considerations in sufficient detail to satisfy the needs of a programmer who is new to computers in general.

## GENERAL DESCRIPTION

The DDP-124, designed and manufactured by Computer Control Company, is a compact and versatile general-purpose computer intended for use in a wide range of data processing applications, including real-time control and scientific computations. Computer operations are performed in the parallel binary mode with 24-bit, single-address instructions, and 24-bit data words. The program is stored in a magnetic core memory, which has a cycle time of 1.75 μsec. The computer features a powerful instruction repertoire, modular design, a wide range of options, and an instruction trap feature providing for program compatibility with the DDP-224.

### Principles of Operation

The DDP-124 General-Purpose Computer performs computations under control of a program, stored in the computer memory. The program consists of a series of commands or instructions. Groups of instructions that perform distinct tasks within the program are called routines and subroutines.

The instructions are actually binary numbers or words and are stored in memory. The central processing unit of the DDP-124 is capable of fetching these words from memory, and decoding them to develop control signals which cause the instruction to be executed. The control signals cause such activities as the addition of specified binary data words

stored in the computer memory, or the starting or stopping of peripheral devices (typewriter, line printer, etc.) connected to the computer. The fetching and execution of instructions make up the basic two-step rhythm of computer operation.

Decoding of the binary digits (bits) within each instruction word is accomplished by the computer through the use of the logical operators, AND, OR, and NOT. The computer can perform only logical operations. Thus, more complex computations (add, multiply, etc.) are derived as extensions of the basic operations, by means of logical design.

Electronic registers are used for storage of binary words in the DDP-124. Each register consists of one or more flip-flops. Each flip-flop is an electronic switch that indicates by its on or off state the presence of a ONE or ZERO, respectively. Small electromagnets are also used as bit storage devices and indicate by their state of polarization the presence of a ZERO or ONE. The basic DDP-124 memory module may be thought of as containing 8192 24-bit registers, which are used to store the binary instruction words and binary data words that make up the program. Each of these registers is assigned an address, so that they may be referred to by instruction words.

A control panel provides the necessary controls for loading the computer memory with the computer program. The coded information comprising the program may be entered into memory directly through the control panel or from a peripheral device (punched-card reader, typewriter, or magnetic-tape unit) connected to the computer.

In addition to decision-making and computation, the command structure of the DDP-124 provides for sensing, responding to, and controlling the operation of displays, controls, and peripheral devices connected to the central processor. These input/output functions enable the computer to act as the control element in a real-time control system, to use auxiliary memories, and to communicate with human operators through a wide variety of transducers. Section III of this manual contains descriptions of the input/output facilities on the DDP-124.

## Command and Data Words

Formats for the binary command or instruction words and the fixed-point binary data words used on the DDP-124 are shown in Figure 1-1. Both command and data words are 24 bits in length.

The first bit of the data word is the sign bit. If this bit is ZERO, the data word is positive; if this bit is ONE, the data word is negative. The remaining 23 bits of the data word carry the value of the word and are called the magnitude bits. With 23 magnitude bits, the maximum number of different data values is $2^{23}$ or 8,388,608. Since all values can either be positive or negative, the total number of possible coded representations becomes 16,777,216.

The command word specifies the operation that is to be performed, and the memory address of the data to be used in the operation. The operation code is contained in bits 4 through 9 of the command word and is represented by two octal integers, or by a 3-letter mnemonic. Bits 10 through 24 generally specify the memory address of the data or "operand" to be used.

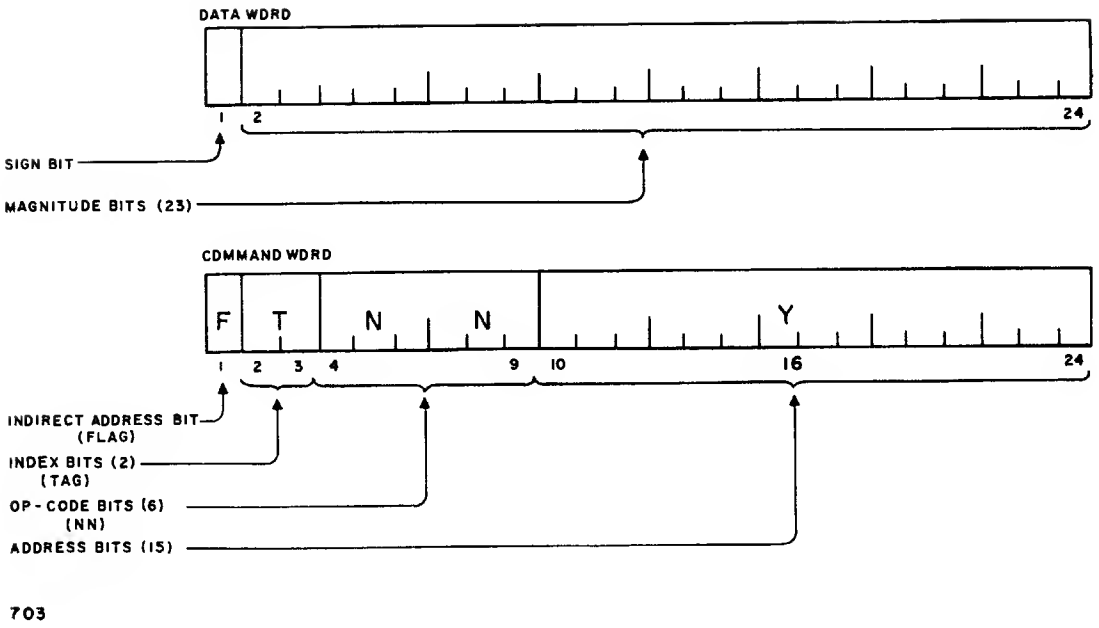



Figure 1-1. Data and Command Word Formats

In addition to the OP-code and address fields in the command word, there are address modification bits: two index bits that specify the index register, if any, that will be used with the operation, and an indirect addressing bit which indicates that the indirect address mode will be used. Bit 1 is for indirect address specification, and bits 2 and 3 are the index bits.

Section II of this manual contains a complete discussion of address modification and other functions of the command word.

## Floating-Point Data Format

Floating-point numbers are expressed as a signed binary fraction (mantissa), multiplied by two raised to an integral power (the exponent). The following table shows several decimal numbers and their equivalent floating-point binary representations (note that the mantissa is always normalized; that is, written so that a significant bit appears immediately to the right of the binary point).

| Decimal Number | Binary Floating-Point Representation | |
|---|---|---|
| | Binary Fraction (mantissa) | Multiplier |
| 4.0 | +0.1 | $2^3$ |
| 0.375 | +0.11 | $2^{-1}$ |
| -17.5 | -0.100011 | $2^5$ |

Floating-point numbers consist of a 9-bit binary exponent and a 23-bit mantissa and sign bit, and are stored in two consecutive locations in memory.

The 9-bit exponent is in true two's-complement form. Therefore, the permissible range of binary exponents is between -256 through +255, as shown in the following table.

| Exponent | Power of 2 Represented |
|---|---|
| 011 111 111 | +255 |
| 011 111 110 | +254 |
| ' | ' |
| ' | ' |
| ' | ' |
| 000 000 010 | +2 |
| 000 000 001 | +1 |
| 000 000 000 | 0 |
| 111 111 111 | -1 |
| 111 111 110 | -2 |
| 111 111 101 | -3 |
| ' | ' |
| ' | ' |
| ' | ' |
| 100 000 001 | -255 |
| 100 000 000 | -256 |

## DDP-124 Block Diagram

Figure 1-2 is a block diagram of the DDP-124, showing the data storage registers and data transfer paths. Storage capacities for all registers are indicated. The logical units illustrated may be combined into four primary functional subdivisions (arithmetic, control, memory, and input/output), as discussed later in this section.

The DDP-124 operates in a relatively typical manner to fetch and interpret each instruction. The contents of the program register indicates the address of the next instruction to be fetched, and are transferred to the memory address register (MAR) to start the process. On a signal from the clock matrix, the memory address register causes the contents of the specified memory location to be transferred to the memory information register (MIR). Bits 1 through 9 of the instruction are transferred from the MIR to the OP-code register for decoding. Bits 1 through 24 are transferred to the Z-register, and the fetch cycle is concluded.

Bits 2 and 3 of the instruction word (the TAG) are now examined for the presence of index bits. If indexing is indicated, the contents of the specified index register are added to bits 10 through 24 of the Z-register, and the sum is present at the output of the adder. If indexing is not specified, bits 10 through 24 of Z are made available at the output of the adder.

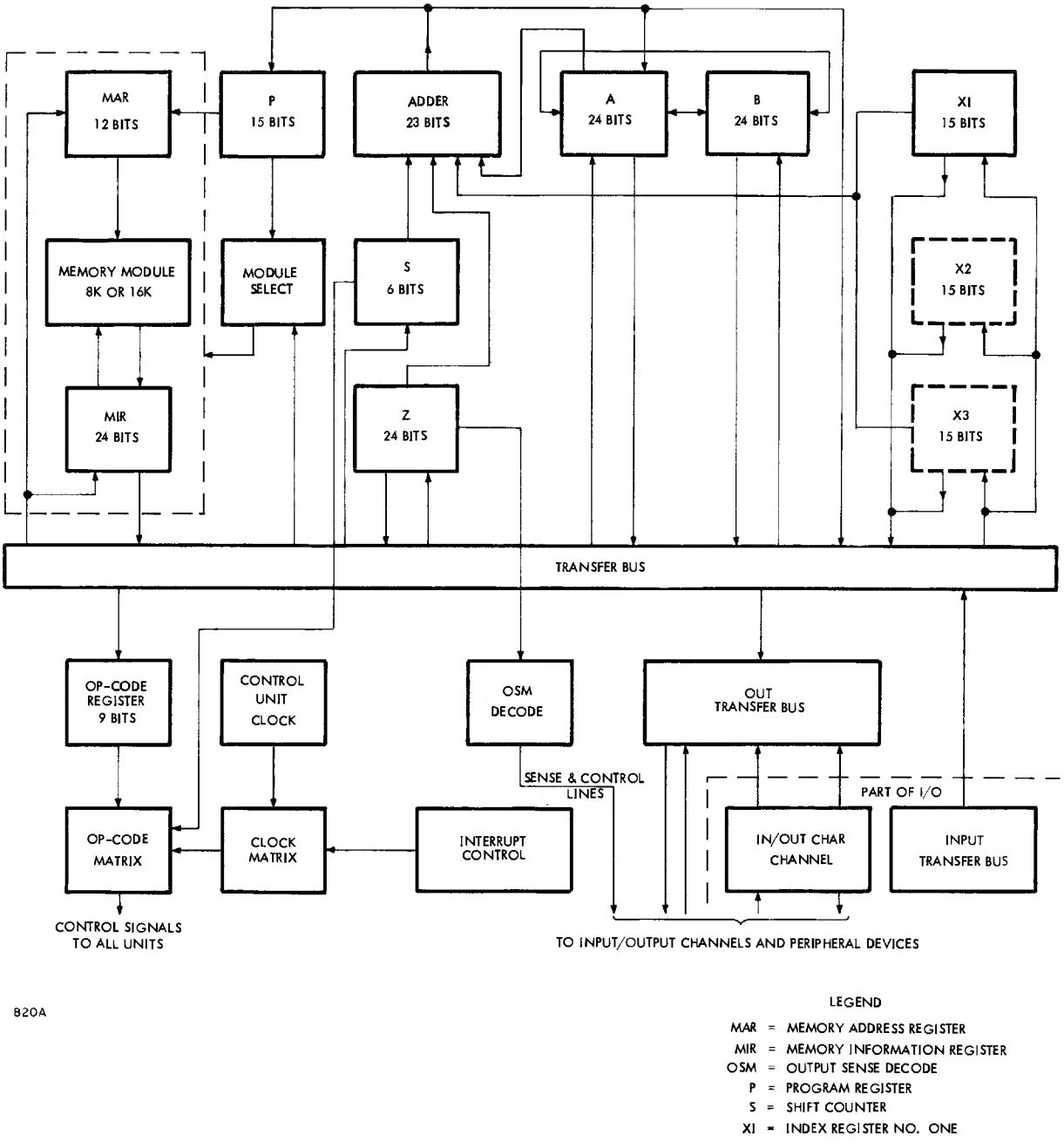



Figure 1-2. DDP-124 Block Diagram

Bit 1 is now checked for the presence of an indirect address bit (the FLAG). If a FLAG is present, the adder output is transferred to the MAR, and the addressed word is brought from memory into the MIR. The execute cycle is again initialized, except that the original code (bits 4 through 9) remains in the OP-code register and retains control of the computer. When no FLAG is found in bit 1, exit from this loop is made, and the operation code is decoded and executed.

## DETAILS OF FUNCTIONAL UNITS

The following discussions cover the main units of the computer and provide the programmer with the information he needs concerning the makeup and capability of the DDP-124.

### Arithmetic Unit

The arithmetic unit contains all of the arithmetic registers in the computer. The contents of the arithmetic registers are visually indicated on the control panel, and each bit content of the A, O, P, and Z registers may be set to a ONE by depressing the pushbutton at the associated bit position on the control panel. Each register for those that can be set from the control panel may be completely reset by depressing the RESET pushbutton adjacent to the register display. The remaining registers are reset by depressing the master clear pushbutton.

A-register. A 24-bit register that is the primary arithmetic and logic register of the computer.

B-register. A 24-bit register that is an extension of the A-register for certain operations including multiplication, division, and certain shifting operations, etc.

Z-register. A 24-bit register that receives its information from the memory. After a command fetch, its 15 least significant bits contain the operand address prior to modification, or in some cases, these 15 bits contain the operand. The nine most significant bits contain the one indirect address bit, the two index bits, and the six OP-code bits. After an operand fetch, the Z-register may contain the operand and then is considered part of the arithmetic unit.

Transfer bus. A gating structure to which the computer registers are connected both as to source and destination. In this way, transfer of information from one register to another can take place by the simultaneous gating of the proper paths from one register to the bus and from the bus to the other register.

Index Register. One index register is standard on the DDP-124 with room for two more when necessary. When a 15-bit index register is specified by an instruction, its contents are added to the address of the instruction, and the result becomes a new address. Display is provided on the control panel.

## Control Unit

The control unit provides the timing and control necessary for the execution of the computer programs. The control unit consists of the following subunits.

Program register (P-register)
OP-code register (O-register)
OP-code matrix (OCM)
Shift counter (S-register)
Transfer bus
Control unit clock
Clock matrix

Program Register. The program register (also a counter) contains the memory location of the next instruction to be performed. Normally its contents are incremented by ONE each time a new command is fetched from memory. In case of a jump, the program counter will be loaded with the memory location to which the program is to jump. The P-register contains 15 bits, is displayed, and can be manually controlled on the computer control panel.

OP-Code Register. The OP-code register contains the 6-bit operation code for the instruction being performed, two index bits, and the indirect address bit for a total of nine bits. It is displayed and manually controlled on the control panel.

OP-Code Matrix. This is a gating structure which decodes the 6-bit OP-code into control signals.

Shift Counter. The 6-bit shift counter is used with instructions that require shifting. It is displayed on the maintenance panel. The shift counter cannot be directly manually controlled, although the single step pushbutton on the maintenance panel will cause stepping of the shift counter during execution of shift instructions.

Control Clock. The control clock is the generator of the timing pulses needed for the various operation sequences of the command executions. Contents are displayed on the maintenance panel. When the CLOCK switch is OFF the clock pulses can be manually controlled by the START pushbutton. Both switches are on the maintenance panel.

Clock Matrix. In the clock matrix, the control unit clock signals are gated with the outputs from the OP-code matrix and supply the correct sequence of pulses and signals for each command to be executed.

## Memory Unit

The DDP-124 computer is available with 8192-word and a 16, 384-word memory module options. The memory may be expanded using these options to a maximum of 32, 768 words, but a maximum of 2 memory modules may be present in a system. Each module contains a memory address register (MAR) and a memory information register (MIR), and may be internally connected to the computer transfer bus or accessed internally from a DMA input/output processor.

### Input/Output Facilities

A standard input/output character buffer and an output bus are provided on the DDP-124 for the transfer of data between the computer and peripheral devices. A wide variety of additional data channels are available as options. Standard sense and control lines have been provided for enabling communication between the computer and its peripheral devices. Input/output facilities are described in detail in Section III of this manual.

## DDP-124 SPECIFICATIONS

### Type

Binary, core memory, parallel, single-address with indexing, and indirect addressing.

### Word Length

24 bits; sign/magnitude code

### Speed (Including Instruction and Operand Access)

| | |
|---|---|
| Add | 3.5 μsec |
| Multiply | 14 μsec (avg) |
| Divide | 22 μsec |
| I/O word transfer | 1.75 μsec |
| I/O block transfer | up to 325,000 words per second |

### Memory

8192 words or 16,384-word modules expandable to 32,768 words; all words directly addressable; ferrite core; non-volatile storage. 1.75 μsec cycle time.

### Input/Output

Input/output character buffers; 24-bit parallel input/output bus. A large number and variety of I/O channels can be added. Program controlled input/output, or automatic interrupt (optional) for any input/output channels. Up to 64 control pulses to peripheral equipment; up to 64 sense input from external sources.

### Signal Levels

Zero volt for logical ZERO; +6 volts for logical ONE.

### Manual Controls

Manual data entry and display of registers; operational controls; maintenance panel for detection of system malfunction.

### Software

FORTRAN - FORTRAN compiler with Boolean statements, capable of operating with basic DDP-124 system (8K memory, typewriter, paper-tape).

DAP2 - Symbolic Assembly Program, relocatable or absolute object output; subroutine linkage, allowing FORTRAN compatibility.

DIP - Scientific Interpretive Programming System

Installation

| | |
|---|---|
| Power: | 3500 watts, single phase, 115 ±10v, 60 cps |
| Operating temperature range: | 10°C to 40°C |
| Humidity: | 0 to 90% |

Requires no air conditioning, wiring, subflooring or other special installation preparation.

Power

Regulated power supplies are included in the DDP-124; no additional regulation is required if input power is within the stated specifications. Overall supply voltage variations due to worst case combinations, input-line voltage changes, dc load regulation, dynamic load regulation, ripple, long term drift, etc., are less than 2% (well within the ±10% circuit tolerances).

# SECTION II
# DDP-124 INSTRUCTIONS

## GENERAL

This section contains a detailed description of each of the DDP-124 instructions. Information presented for each instruction includes the following:

The 3-letter mnemonic that will be interpreted by DAP2, the DDP-124 Assembly Program, and the machine language translator in FORTRAN

The definition of the mnemonic

The instruction execution time (including memory access)

The instruction format showing the octal code

A description of the effect of the instruction

A discussion of the effects of tagging the instruction with an index register

A discussion of the effects of flagging the instruction with the indirect address bit

A discussion of the indicators affected by the instruction

A diagram showing the step-by-step execution of the instruction

### Instruction Mnemonics

Each of the DDP-124 instructions has been assigned an alphabetic code (mnemonic) making the instructions easier to memorize and recall. In the assignment of mnemonic codes, certain considerations have been generally followed.

1. All codes are composed of three alphabetic characters.
2. When an instruction involves the loading of some register from memory, the code generally starts with LD.
3. When an instruction involves the storing of some register into memory, the code generally starts with ST.
4. When an instruction involves a skipping operation, the code starts with SK.
5. When an instruction involves a jumping operation, the code starts with J.
6. Whenever possible, the name of the register involved is part of the code (A for the A-register, B for the B-register, and X for an index register).

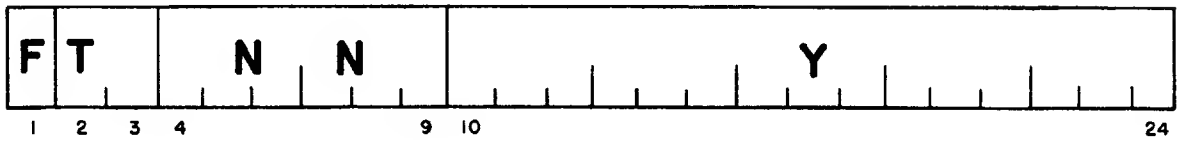
### Execution Time

The figure noted to the right of the instruction is the time in microseconds required for the execution of the instruction, including instruction and operand fetching, and indexing.

The time noted does not provide for indirect addressing. For each level of indirect addressing, 1.75 μsec should be added to the basic execution time.

## Format

The format for DDP-124 instruction is as follows.




F is the FLAG portion

0 indicates that the instruction is not flagged for indirect addressing.

1 indicates that the instruction is flagged for indirect addressing.

T is the TAG portion

0 indicates that the instruction is not tagged for indexing.

1 indicates that the instruction is tagged for indexing with index register 1.

2 indicates that the instruction is tagged for indexing with index register 2 (optional).

3 indicates that the instruction is tagged for indexing with index register 3 (optional).

NN is the octal representation of the binary instruction or operation code.

Y is the effective address specified by the instruction (normally, the effective address of the operand).

## Effective Address

The address field or address portion of the instruction word consists of bits 10 through 24. If the instruction word contains no FLAG or TAG (i.e., no indirect address bit and no index bits), the contents of the address field of the instruction word specifies the address at which the operand is located. The address at which the operand is located is called the effective address of the operand.

When a FLAG and/or TAG is present in the instruction word, the effective address of the operand is generally not equal to the contents of the address portion of the instruction word. In these cases, the effective address of the operand is determined as follows.

1. When the instruction word contains a FLAG and no TAG, the contents of the address portion of the instruction word specifies a location in memory. The contents of the address portion of that word is the effective address of the operand if the word contains no FLAG or TAG bits.

2. When the instruction word contains a TAG and no FLAG, the contents of the specified index register is added to the contents of the address portion of the instruction word. The resultant sum is the effective address of the operand. More complex methods for obtaining an effective address are possible, and are discussed below, in the paragraphs on indirect addressing.

In the descriptions of some of the instructions, the effective address is not always used to address a word in memory. For example, the effective address for shift instructions is the number of positions that the contents of the registers is to be shifted.

## Indexed Addressing

Address modification, by means of an index register, may be specified on most of the DDP-124 instructions. This is accomplished by entering the number of the index register (1, 2, or 3) into the TAG portion of the instruction word.

NOTE

All DDP-124s have at least one index register, and may have three. If index register 2 or 3 is specified on a computer with only index register 1, the instruction will be executed as though no index register had been specified. All index registers contain 15 bits.

When an instruction is tagged, the contents of the specified index register is added to the address portion of the instruction, forming a 15-bit sum. This sum becomes the new address. When subtraction is desired, the number to be subtracted should be placed into the index register in twos-complement form.

## Indirect Addressing

All DDP-124 instructions can be flagged for indirect addressing. When a FLAG is present, the computer proceeds as follows.

Step 1. The TAG is examined; if indexing is specified, an indexed address is formed in the normal manner (sum of the specified index register and the address portion of the instruction word). If no index register is specified, the computer proceeds to step 2 without altering the address field.

Step 2. The word in the memory location specified by the address in step 1 is accessed.

Step 3. When this word has a FLAG, the computer goes back to step 1; when it does not have a FLAG, the computer goes to step 4.

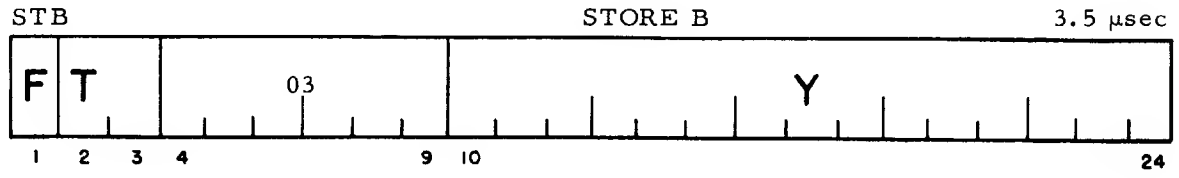
Step 4. The TAG field and address field of this word replace the TAG field and address field of the initial instruction.

Step 5. When the TAG field specifies indexing, the sum of the address field and the index register form the effective address. When no TAG is specified, the address field is the effective address.

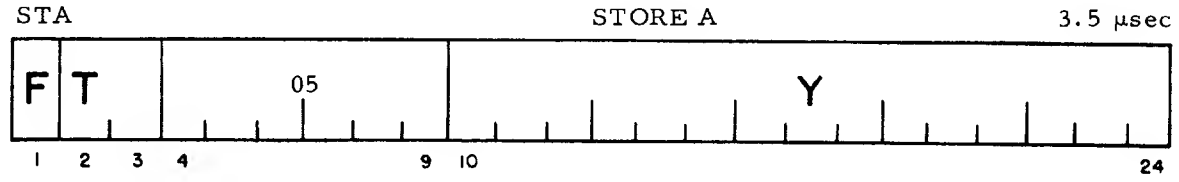
Step 6. The effective instruction has now been formed and is executed.

This process is followed for every DDP-124 instruction except those which cannot be indexed. Regardless of the effects of indirect addressing, 1.9 μsec are required for each level.

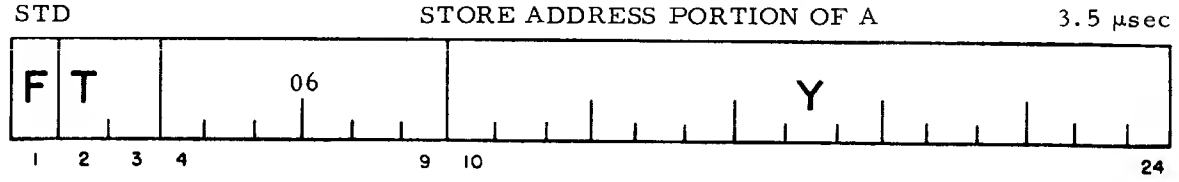
## LOAD AND STORE INSTRUCTIONS




DESCRIPTION: The contents of the B-register, bits 1 through 24, replace the contents of Y, bits 1 through 24. The contents of the B-register, bits 1 through 24, are unchanged.
TAG: Normal
FLAG: Normal
INDICATORS: None

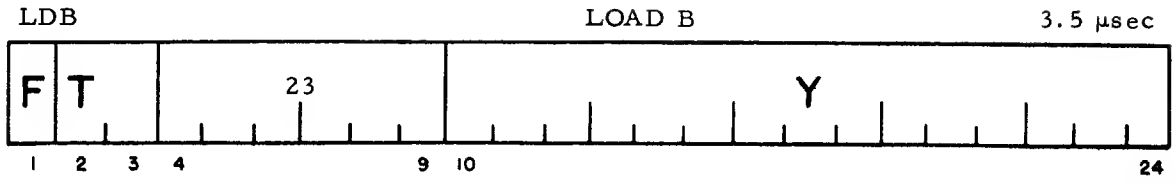



DESCRIPTION: The contents of the A-register, bits 1 through 24, replace the contents of Y, bits 1 through 24. The contents of the A-register, bits 1 through 24, are unchanged.
TAG: Normal
FLAG: Normal
INDICATORS: None

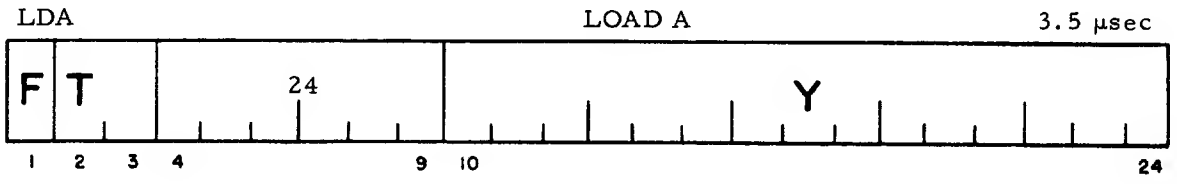



DESCRIPTION: The contents of the A-register, bits 10 through 24, replace the contents of Y, bits 10 through 24. The contents of the A-register, bits 1 through 24, and the command portion of Y, bits 1 through 9, are unchanged.

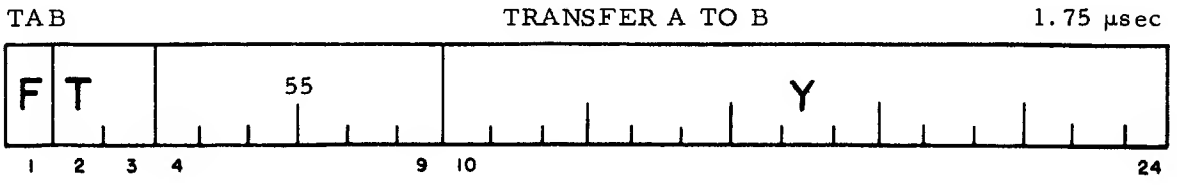
TAG: Normal
FLAG: Normal
INDICATORS: None




DESCRIPTION: The contents of Y, bits 1 through 24, replace the contents of the B-register, bits 1 through 24. The contents of Y, bits 1 through 24, are unchanged.
TAG: Normal
FLAG: Normal
INDICATORS: None

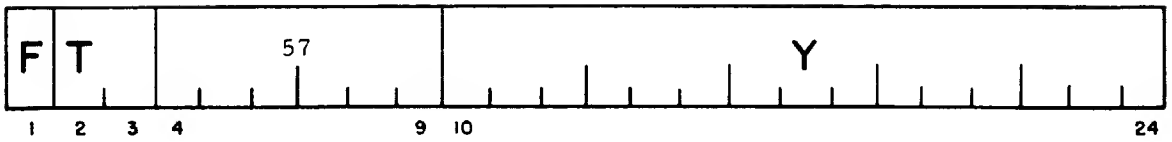



DESCRIPTION: The contents of Y, bits 1 through 24, replace the contents of the A-register, bits 1 through 24. The contents of Y, bits 1 through 24, are unchanged.
TAG: Normal
FLAG: Normal
INDICATORS: None




DESCRIPTION: The contents of the A-register, bits 1 through 24, replace the contents of the B-register, bits 1 through 24. The contents of the A-register, bits 1 through 24, are unchanged. The address portion of this instruction, bits 10 through 24, is not interpreted unless indirect addressing is specified.
TAG: Does not affect the instruction
FLAG: Does not affect the instruction, except to delay its execution.
INDICATORS: None

IAB INTERCHANGE A AND B 1.75 μsec



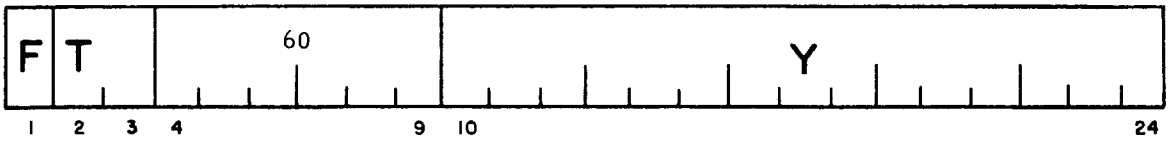

DESCRIPTION: The contents of the A-register, bits 1 through 24, and the B-register, bits 1 through 24, are interchanged. The address portion of this instruction, bits 10 through 24, is not interpreted unless indirect addressing is specified.

TAG: Does not affect the instruction

FLAG: Does not affect the instruction, except to delay its execution.

INDICATORS: None

CRA CLEAR A 1.75 μsec




DESCRIPTION: The contents of the A-register, bits 1 through 24, are set to ZEROs. The address portion of this instruction, bits 10 through 24, is not interpreted unless indirect addressing is specified.

TAG: Does not affect the instruction

FLAG: Does not affect the instruction, except to delay its execution.

INDICATORS: None

## ARITHMETIC INSTRUCTIONS

ADD ADD 3.5 μsec


DESCRIPTION: The contents of Y, sign and bits 2 through 24, are algebraically added to the contents of the A-register, sign and bits 2 through 24, and the resultant sum replaces the contents of the A-register, sign and bits 2 through 24. If the magnitude of the sum, bits 2 through 24, is ZERO, the initial sign of the A-register is unchanged. The contents of Y, sign and bits 2 through 24, are unchanged.

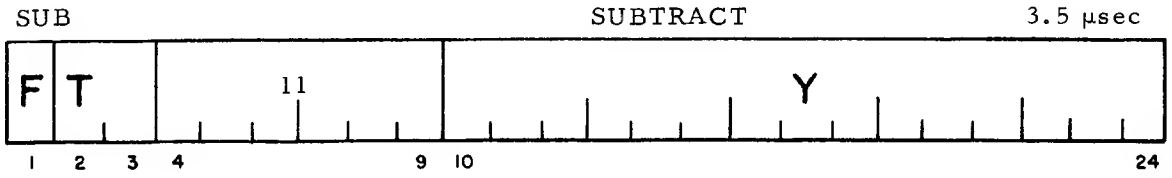
TAG: Normal

FLAG: Normal

INDICATORS: If overflow occurs, the OVERFLOW indicator will be set and the computer will proceed to the next sequential instruction. The resultant sign in the A-register will be the algebraic sign of the result. The magnitude of the result will be modulo $2^{23}$.

NOTE

Overflow is not possible if the initial sign of the contents of the A-register and the sign of the contents of Y are not alike.




DESCRIPTION: The contents of Y, sign and bits 2 through 24, are algebraically subtracted from the contents of the A-register, sign and bits 2 through 24, and the resultant difference replaces the contents of the A-register, sign and bits 2 through 24. If the magnitude of the difference, bits 2 through 24, is ZERO, the initial sign of the A-register is unchanged. The contents of Y, sign and bits 2 through 24, are unchanged.

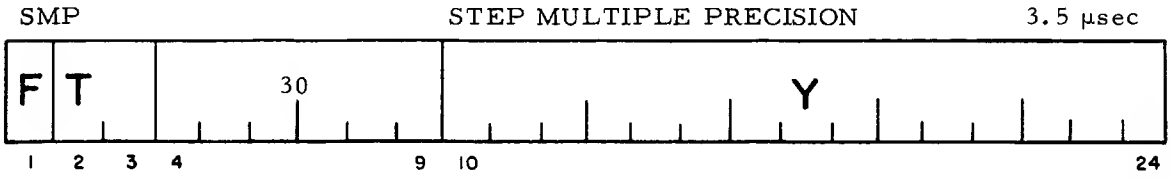
TAG: Normal

FLAG: Normal

INDICATORS: If overflow occurs, the OVERFLOW indicator will be set and the computer will proceed to the next sequential instruction. The resultant sign in the A-register will be the algebraic sign of the result. The magnitude of the result will be modulo $2^{23}$.

NOTE

Overflow is not possible if the initial sign of the contents of the A-register and the sign of the contents of Y are alike.




DESCRIPTION: The SMP instruction may be used to add or subtract two numbers, each having a length greater than 23 bits (multiple precision). Each multiple precision number must be stored in two or more computer words (not necessarily consecutive), and the signs of each of the words comprising a multiple precision number must agree.

The arithmetic operation to be performed by the SMP instructions must be established by adding (ADD) or subtracting (SUB) the words containing the most significant portions of the numbers to be operated on. When the result of this ADD or SUB is zero, the

multiple precision subroutine must be coded so that the next most significant portions of the numbers are added or subtracted and the result checked. This procedure must be repeated until a non-zero result is obtained (if the sums or differences of all parts of the numbers are ZERO, the sum or difference of the two multiple precision numbers is ZERO). When a non-zero result is obtained, the SMP instruction is then used to add or subtract the two multiple precision numbers in the following manner: the least significant portion of number m must be placed in the A-register (LDA). An SMP instruction addressing the least significant portion of number n is executed and the results replace the contents of the A-register, sign and bits 2 through 24. The portion of number n addressed by the SMP instruction is unchanged. The least significant result in the A-register must be stored (STA) in the word reserved for the least significant portion of the answer. If a carry is generated out of bit position 2 of the A-register, or a borrow was required, an internal flag is set and the carry or borrow is accounted for when the next SMP instruction is executed.

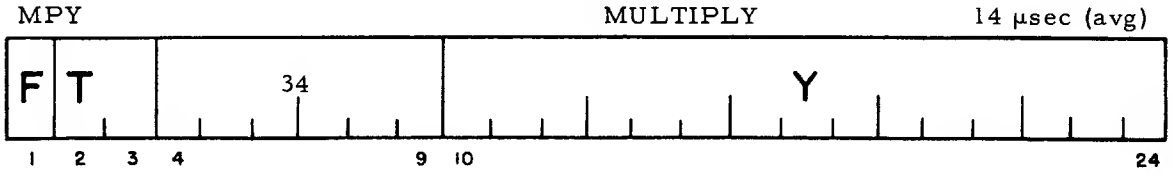
An SMP instruction must be executed in the above manner for each word of the multiple precision numbers, taken in order of increasing significance up to and including the most significant portions. An example of a multiple precision addition subroutine is shown in Section IV.

Any instruction except for JRT can be executed between the initial ADD or SUB and the first SMP instruction.

TAG: Normal

FLAG: Normal

INDICATORS: The OVERFLOW indicator is reset at the beginning of each SMP instruction; if overflow occurs during the execution of the final SMP, the OVERFLOW indicator will remain ON at the conclusion of the instruction.

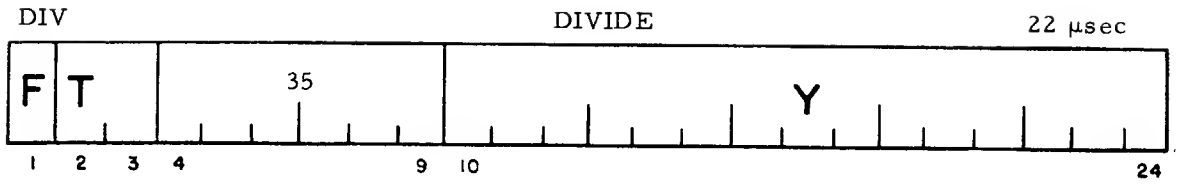



DESCRIPTION: The contents of the B-register, sign and bits 2 through 24, are multiplied by the contents of Y, sign and bits 2 through 24. The most significant bits of the 46-bit product replace the contents of the A-register, bits 2 through 24. The least significant 23 bits replace the contents of the B-register, bits 2 through 24. The signs of the A- and B-registers are set to the algebraic sign of the product. The contents of the A-register, sign and bits 2 through 24, are set to ZERO at the start of this instruction; the contents of Y, sign and bits 2 through 24 are unchanged.

TAG: Normal

FLAG: Normal

INDICATORS: None

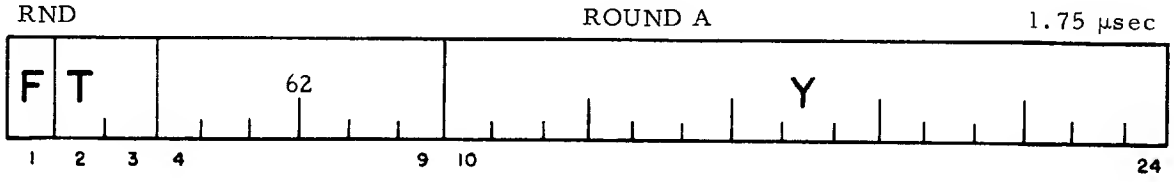



DESCRIPTION: The contents of Y, sign and bits 2 through 24 (the divisor), are divided into the contents of both the A-register and the B-register, sign and bits 2 through 24 (the double-length dividend). The 23-bit quotient replaces the contents of the B-register, bits 2 through 24. The 23-bit remainder replaces the contents of the A-register, bits 2-24. The sign of the B-register is set to the algebraic sign of the quotient; the sign of the A-register is set to the initial sign of the dividend. The contents of Y, sign and bits 2 through 24 are unchanged.

TAG: Normal

FLAG: Normal

INDICATORS: If the initial magnitude of the A-register, bits 2 through 24, is equal to or greater than the magnitude of the contents of Y, bits 2 through 24, the improper divide indicator is set, and the computer will proceed to the next sequential instruction.

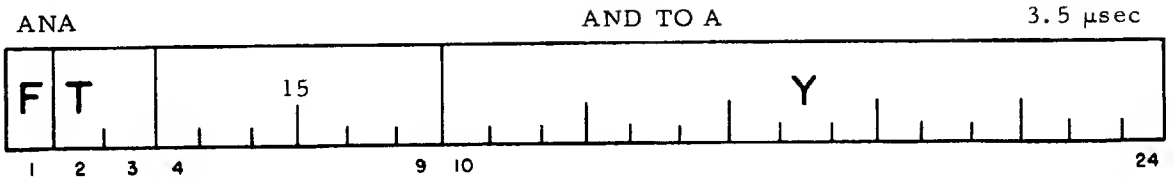



DESCRIPTION: The contents of the A-register, bits 2 through 24, are incremented by one if bit 2 in the B-register is a ONE. The contents of the A-register, bits 2 through 24, are unchanged if bit 2 in the B-register is a ZERO. The address portion of this instruction, bits 10 through 24, is not interpreted unless indirect addressing is specified. The contents of the B-register, bits 1 through 24, are unchanged.

TAG: Does not affect the instruction

FLAG: Does not affect the instruction, except to delay its execution.

INDICATORS: If overflow occurs, the overflow indicator will be set and the computer will proceed to the next sequential instruction. The resultant sign in the A-register will not be changed. The magnitude of the result will be zero.

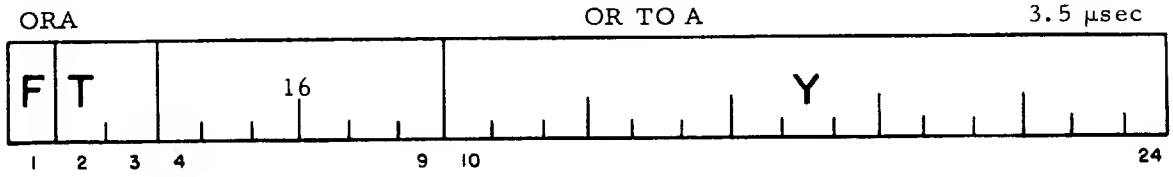



DESCRIPTION: The logical product of the contents of the A-register, bits 1 through 24, and the contents of Y, bits 1 through 24, is formed, and the result replaces the contents of the A-register, bits 1 through 24. For each ZERO in the contents of Y, a ZERO is placed in the corresponding bit position in the A-register. For each ONE in the contents of Y, the corresponding bit position in the A-register is unchanged. The contents of Y, bits 1 through 24, are unchanged.

TAG: Normal

FLAG: Normal

INDICATORS: None



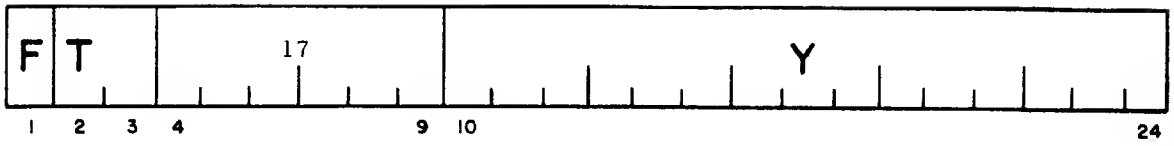

DESCRIPTION: The logical sum of the contents of the A-register, bits 1 through 24, and the contents of Y, bits 1 through 24, is formed and the result replaces the contents of the A-register, bits 1 through 24. For each ONE in the contents of Y, a ONE is placed in the corresponding bit position in the A-register. For each ZERO in the contents of Y, the corresponding bit position in the A-register is unchanged. The contents of Y, bits 1 through 24, are unchanged.

TAG: Normal

FLAG: Normal

INDICATORS: None

ERA EXCLUSIVE OR TO A 3.5 μsec




DESCRIPTION: The logical difference of the contents of the A-register, bits 1 through 24, and the contents of Y, bits 1 through 24, is formed, and the result replaces the contents of the A-register, bits 1 through 24. For each ONE in the contents of Y, the corresponding bit position in the A-register is complemented. For each ZERO in the contents of Y, the corresponding bit position in the A-register is unchanged. The contents of Y, bits 1 through 24, are unchanged.

TAG: Normal

FLAG: Normal

INDICATORS: None

## SHIFT INSTRUCTIONS

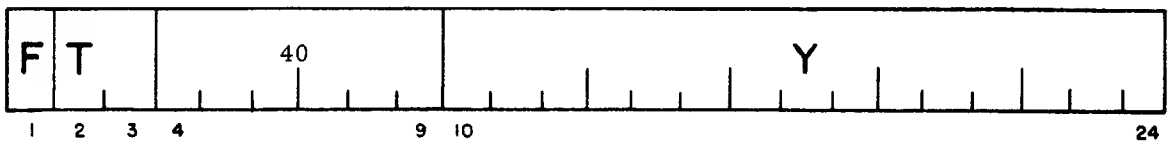
### Timing

A formula for calculating execution time is included in the discussion for each shift instruction. These formulas include the variable x, which assumes one of the following values, depending upon the number of shift steps, n, required to execute the instruction:

$$\text{If } n \geq 4,\ x = 0.35\ (n)\ \mu\text{sec}$$

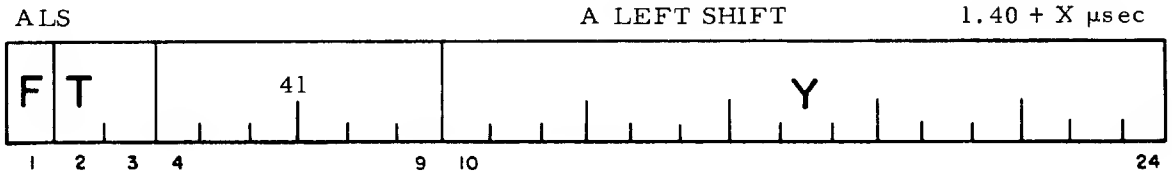
$$\text{If } n < 4,\ x = 1.05\ \mu\text{sec}$$

ARS A RIGHT SHIFT 1.40 + X μsec

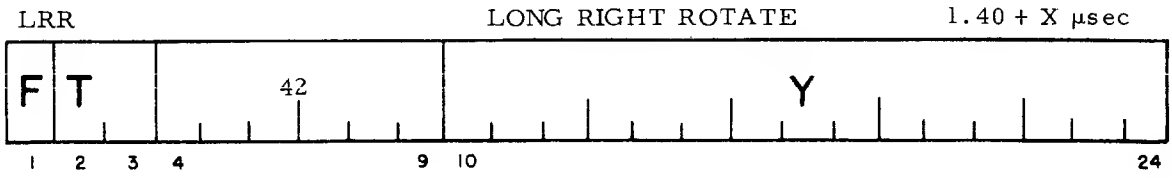



DESCRIPTION: The contents of the A-register, bits 2 through 24, are shifted to the right the number of positions specified by the six least significant address bits of this instruction, bits 19 through 24. The sign of the A-register is not shifted and is unchanged. ZEROs are shifted into the vacated position next to the sign of the A-register, bit 2. Bits shifted out of the low-order position of the A-register, bit 24, are lost. A maximum shift of 63 positions is possible; however, if a shift larger than 22 occurs, bits 2 through 24 of the A-register will be ZEROs.

TAG: The contents of the specified index register will be added to the address of this instruction, bits 10 through 24; however, only the low-order six bits of the resultant sum will be interpreted for the length of the shift.
FLAG: If a flag is specified, the contents of Y, bits 19 through 24, will specify the length of the shift. If a tag is specified in the contents of Y, the length of the shift will be determined as described under tag.
INDICATORS: None




DESCRIPTION: The contents of the A-register, bits 2 through 24, are shifted to the left the number of positions specified by the six least significant address bits of this instruction, bits 19 through 24. The sign of the A-register is not shifted and is unchanged. ZEROs are shifted into the vacated low-order position of the A-register, bit 24. Bits shifted out of the position next to the sign of the A-register, bit 2, are lost (see indicators). A maximum shift of 63 positions is possible; however, if a shift larger than 22 occurs, bits 2 through 24 of the A-register will be ZEROs.
TAG: The contents of the specified index register will be added to the address of this instruction, bits 10 through 24; however, only the low-order six bits of the resultant sum will be interpreted for the length of the shift.
FLAG: If a flag is specified, the contents of Y, bits 19 through 24, will specify the length of the shift. If a tag is specified in the contents of Y, the length of the shift will be determined as described under tag.
INDICATORS: If a ONE is shifted out of bit position 2 of the A-register, the OVERFLOW indicator will be set, and the computer will proceed to the next sequential instruction, but not until all specified shifts have occurred.




DESCRIPTION: The contents of the A-register bits 1 through 24, and the B-register, bits 1 through 24, are treated as a single 48-bit register and are rotated to the right (end-around carry) the number of positions specified by the six least significant address bits of this instruction, bits 19 through 24. Bits shifted out of the low-order position of the A-register,

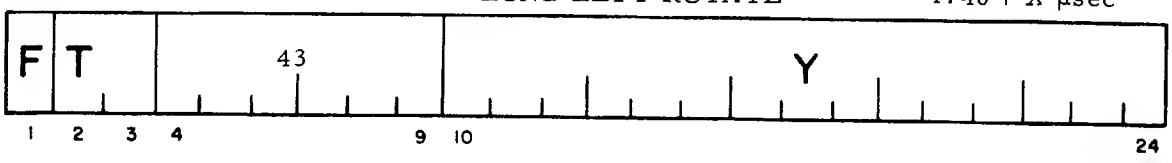
bit 24, enter bit position 1 of the B-register. Bits shifted out of the low-order position of the B-register, bit 24, enter bit position 1 of the A-register. A maximum shift of 63 positions is possible.

TAG: The contents of the specified index register will be added to the address of this instruction, bits 10 through 24; however, only the low-order six bits of the resultant sum will be interpreted for the length of the shift.

FLAG: If a FLAG is specified, the contents of Y, bits 19 through 24, will specify the length of the shift. If a TAG is specified in the contents of Y, the length of the shift will be determined as described under TAG.

INDICATORS: None

LLR LONG LEFT ROTATE 1.40 + X μsec



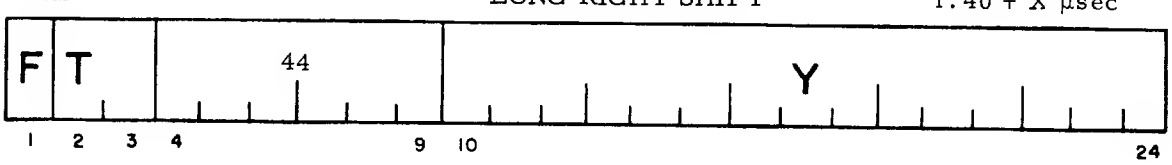

DESCRIPTION: The contents of the A-register, bits 1 through 24, and the B-register, bits 1 through 24, are treated as a single 48-bit register and are rotated to the left (end-around carry) the number of positions specified by the six least significant address bits of this instruction, bits 19 through 24. Bits shifted out of bit position 1 of the A-register, enter the low-order position of the B-register, bit 24. Bits shifted out of bit position 1 of the B-register enter the low-order position of the A-register, bit 24. A maximum shift of 63 positions is possible.

TAG: The contents of the specified index register will be added to the address of this instruction, bits 10 through 24; however, only the low-order six bits of the resultant sum will be interpreted for the length of the shift.

FLAG: If a FLAG is specified, the contents of Y, bits 19 through 24, will specify the length of the shift. If a TAG is specified in the contents of Y, the length of the shift will be determined as described under TAG.

INDICATORS: None

LRS LONG RIGHT SHIFT 1.40 + X μsec



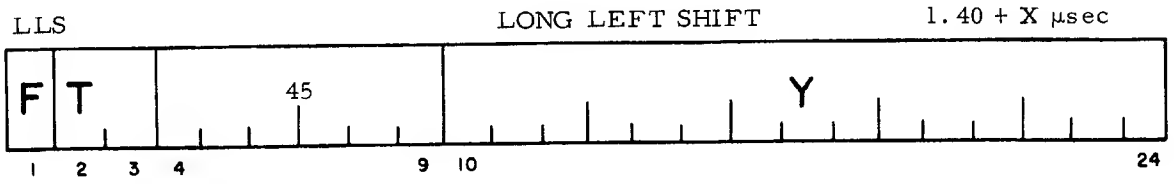

DESCRIPTION: The contents of the A-register, bits 2 through 24, and the B-register, bits 2 through 24, are treated as a single 46-bit register and are shifted to the right the number of positions specified by the six least significant address bits of this instruction, bits 19

through 24. The signs of the A- and B-registers are not shifted; however, the sign of the B-register is made to agree with the sign of the A-register. ZEROs are shifted into the vacated position next to the sign of the A-register, bit 2. Bits shifted out of the low-order position of the A-register, bit 24, enter the position next to the sign of the B-register, bit 2. Bits shifted out of the low-order position of the B-register, bit 24, are lost. A maximum shift of 63 positions is possible; however, if a shift larger than 45 occurs, bits 2 through 24 of the A- and B-registers will be ZEROs.

TAG: The contents of the specified index register will be added to the address of this instruction, bits 10 through 24; however, only the low-order six bits of the resultant sum will be interpreted for the length of the shift.

FLAG: If a FLAG is specified, the contents of Y, bits 19 through 24, will specify the length of the shift. If a TAG is specified in the contents of Y, the length of the shift will be determined as described under TAG.

INDICATORS: None



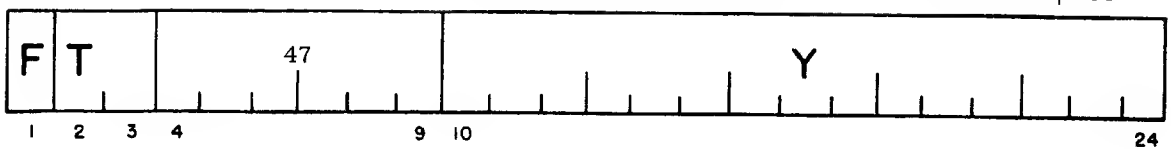

DESCRIPTION: The contents of the A-register, bits 2 through 24, and the contents of the B-register, bits 2 through 24, are treated as a single 46-bit register and are shifted to the left the number of positions specified by the six least significant address bits of this instruction, bits 19 through 24. The signs of the A- and B-registers are not shifted; however, the sign of the A-register is made to agree with the sign of the B-register. ZEROs are shifted into the vacant low-order position of the B-register, bit 24. Bits shifted out of the position next to the sign of the B-register, bit 2, enter the low-order position of the A-register, bit 24. Bits shifted out of the position next to the sign of the A-register, bit 2, are lost (see indicators). A maximum shift of 63 positions is possible; however, if a shift larger than 45 occurs, bits 2 through 24 of the A- and B-registers will be ZERO.

TAG: The contents of the specified index register will be added to the address of this instruction, bits 10 through 24; however, only the low-order six bits of the resultant sum will be interpreted for the length of the shift.

FLAG: If a FLAG is specified, the contents of Y, bits 19 through 24, will specify the length of the shift. If a TAG is specified in the contents of Y, the length of the shift will be determined as described under TAG.

INDICATORS: If a ONE is shifted out of bit position 2 in the A-register, the OVERFLOW indicator will be set and the computer will proceed to the next sequential instruction, but not before the specified number of shifts has occurred.

LGL LOGICAL LEFT SHIFT 1.40 + X μsec




DESCRIPTION: The contents of the A-register, bits 1 through 24, are shifted left the number of positions specified by the six least significant address bits of this instruction, bits 19 through 24. ZEROs are shifted into the vacated low-order position of the A-register, bit 24. Bits shifted out of bit position 1 of the A-register are lost. A maximum shift of 63 positions is possible; however, if a shift larger than 23 occurs, bits 1 through 24 of the A-register will be ZEROs.

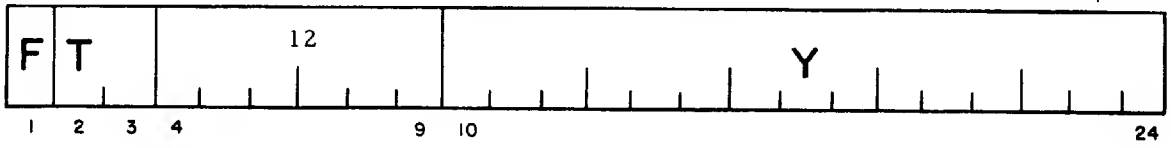
TAG: The contents of the specified index register will be added to the address of this instruction, bits 10 through 24; however, only the six low-order bits of the resultant sum will be interpreted for the length of the shift.

FLAG: If a FLAG is specified, the contents of Y, bits 19 through 24, will specify the length of the shift. If a TAG is specified in the contents of Y, the length of the shift will be determined as described in TAG.

INDICATORS: None

## JUMP AND SKIP INSTRUCTIONS

SKG SKIP IF A GREATER 4.60 μsec




DESCRIPTION: The contents of the A-register, sign and bits 2 through 24, are algebraically compared to the contents of Y, sign and bits 2 through 24. If the value in the A-register is greater than the value in Y, the next sequential instruction is skipped. If the value in the A-register is equal to or less than the value in Y, the computer proceeds to the next sequential instruction. The contents of the A-register, sign and bits 2 through 24, and the contents of Y, sign and bits 2 through 24, are unchanged.

NOTE

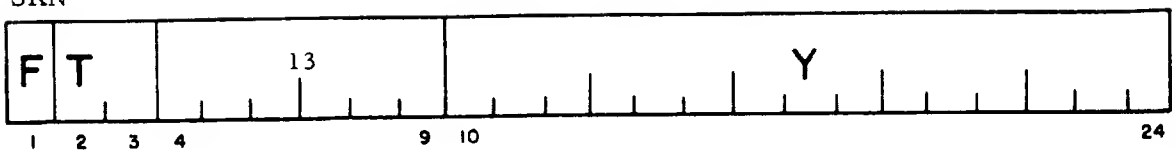
+0 is equal to -0.

TAG: Normal

FLAG: Normal

INDICATORS: None

SKN SKIP IF A NOT EQUAL 4.60 μsec



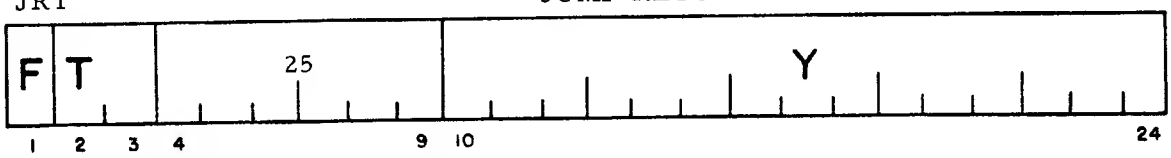

DESCRIPTION: The contents of the A-register, bits 1 through 24, are logically compared to the contents of Y, bits 1 through 24. If the value in the A-register is not equal to the value in Y, the next sequential instruction is skipped. If the value in the A-register is equal to the value in Y, the computer proceeds to the next sequential instruction. The contents of the A-register, bits 1 through 24, and the contents of Y, bits 1 through 24, are unchanged.

TAG: Normal

FLAG: Normal

INDICATORS: None

JRT JUMP RETURN 3.5 μsec



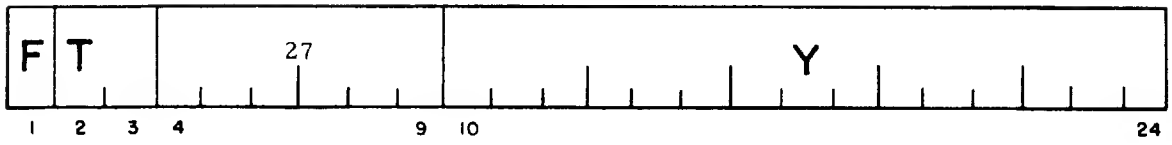

DESCRIPTION: The JRT instruction is used to return program control to an interrupted program. The computer will take its next instruction from the location specified by the address portion of the contents of Y, bits 10 through 24. The contents of Y, bits 1 through 9, are used to restore certain internal flip-flops to their condition at the time the interrupt occurred (see discussion on interrupts for restrictions on Y and details on bits 1 through 9 of the contents of Y). The JRT instruction must be the last instruction executed in an interrupt program. If the standard interrupt is used, this instruction will restore the capability for being interrupted. If priority interrupt (optional) is used, this instruction will return program control to the interrupted priority level. The contents of Y, bits 1 through 24, are unchanged.

TAG: Normal

FLAG: Normal

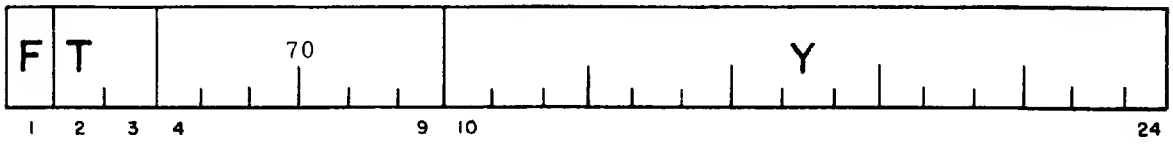
INDICATORS: The INTERRUPT indicator turned OFF.

JST JUMP AND STORE LOCATION 3.5 μsec




DESCRIPTION: The location plus one of this instruction replaces the address portion of the contents of Y, bits 10 through 24. The computer then takes its next instruction from Y + 1 and continues from there. The contents of Y, bits 1 through 9, and Y + 1, bits 1 through 24, are unchanged. This instruction may be used to enter a subroutine.

JPL JUMP IF A PLUS 2.10 μsec



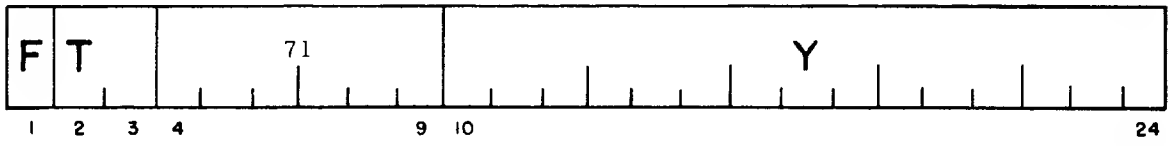

DESCRIPTION: If the sign of the contents of the A-register is positive (ZERO), the computer takes its next instruction from the memory location specified by Y and continues from there. If the sign of the A-register is negative (ONE), the computer proceeds to the next sequential instruction. The contents of the A-register, sign and bits 2 through 24, and the contents of Y, bits 1 through 24 are unchanged. The magnitude portion of the A-register, bits 2 through 24, is not examined by this instruction.

TAG: Normal

FLAG: Normal

INDICATORS: None

JZE JUMP IF A ZERO 2.10 μsec

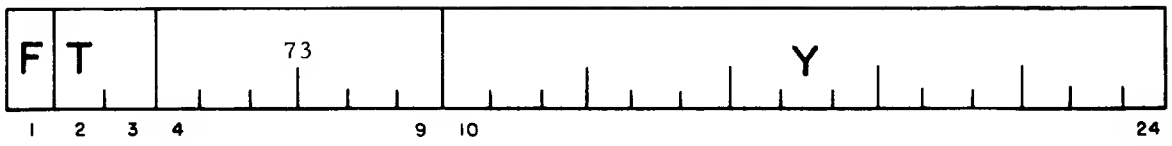



DESCRIPTION: If the magnitude portion of the A-register, bits 2 through 24, is ZERO, the computer takes its next instruction from the memory location specified by Y and continues from there. If any of the bits in the magnitude portion of the A-register are ONEs, the computer will proceed to the next sequential instructions. The contents of the A-register, bits 1 through 24, and the contents of Y, bits 1 through 24, are unchanged. The sign of the A-register is ignored by this instruction.

TAG: Normal

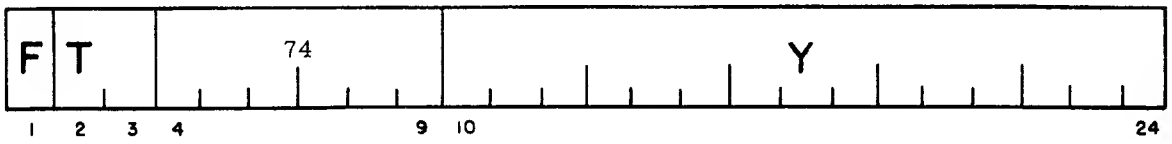
FLAG: Normal
INDICATORS: None

JOF JUMP ON OVERFLOW 2.10 μsec




DESCRIPTION: If the OVERFLOW indicator is set, it will be reset and the computer will take its next instruction from the memory location specified by Y and continue from there. If the OVERFLOW indicator is not set, the computer will proceed to the next sequential instruction. The contents of Y, bits 1 through 24, are unchanged.
TAG: Normal
FLAG: Normal
INDICATORS: The OVERFLOW indicator is turned off.

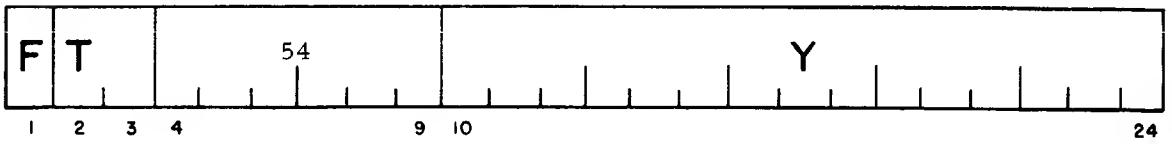
JMP UNCONDITIONAL JUMP 2.10 μsec




DESCRIPTION: The computer takes its next instruction from the memory location specified by Y and continues from there. The contents of Y, bits 1 through 24, are unchanged.
TAG: Normal
FLAG: Normal
INDICATORS: None

INDEX INSTRUCTIONS

ADX ADD TO INDEX 1.75 μsec

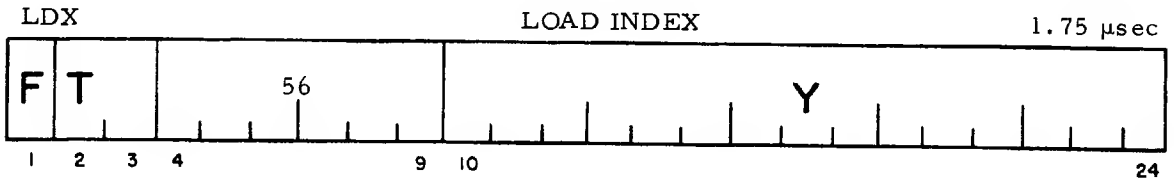



DESCRIPTION: The contents of the address portion of this instruction, bits 10 through 24, are added to the contents of the specified index register and the resultant sum replaces the contents of the specified index register. Overflow of the specified index register is possible, but will be ignored.

TAG: The TAG specifies which index register will be modified. If no TAG is specified (bits 2 and 3 are ZERO), this instruction is treated as an NOP (unless indirect addressing is specified, see below).

FLAG: If this instruction is flagged for indirect addressing, the normal procedure for forming an effective address will be followed and the address portion of the contents of Y, bits 10 through 24, will be added to the index register specified in the TAG portion of Y, bits 2 and 3.

INDICATORS: None

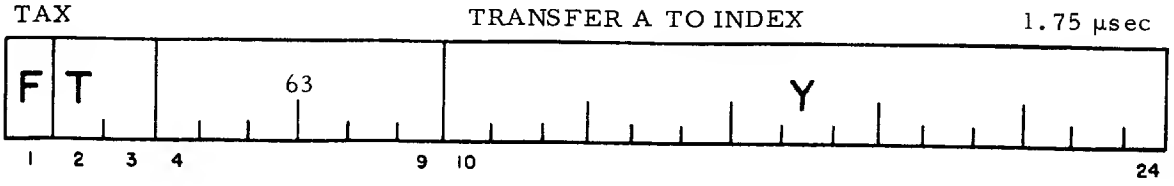



DESCRIPTION: The contents of the address portion of this instruction, bits 10 through 24 replace the contents of the specified index register.

TAG: The TAG specifies which index register will be modified. If no TAG is specified (bits 2 and 3 are ZERO), this instruction is treated as an NOP (unless indirect addressing is specified, see below).

FLAG: If this instruction is flagged for indirect addressing, the normal procedure for forming an effective address will be followed and the address portion of the contents of Y, bits 10 through 24, will replace the contents of the index register specified in the TAG portion of Y, bits 2 and 3.

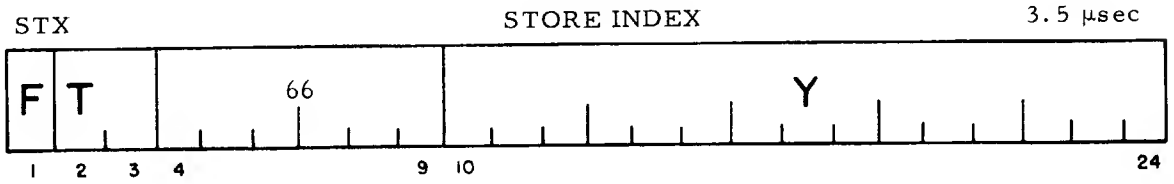
INDICATORS: None



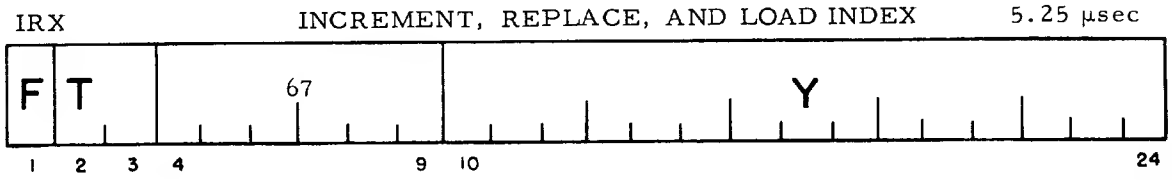

DESCRIPTION: The address portion of the A-register, bits 10 through 24, replaces the contents of the specified index register. The contents of the A-register, bits 1 through 24, are unchanged. The address portion of this instruction, bits 10 through 24, is not interpreted unless indirect addressing is specified.

TAG: The TAG specifies which index register will be modified. If no TAG is specified (bits 2 and 3 are ZERO), this instruction is treated as an NOP (unless indirect addressing is specified, see below).

FLAG: If this instruction is flagged for indirect addressing, the normal procedure for forming an effective address will be followed, and the TAG portion of the contents of Y, bits 2 and 3, specifies which index register will be affected by this instruction.
INDICATORS: None



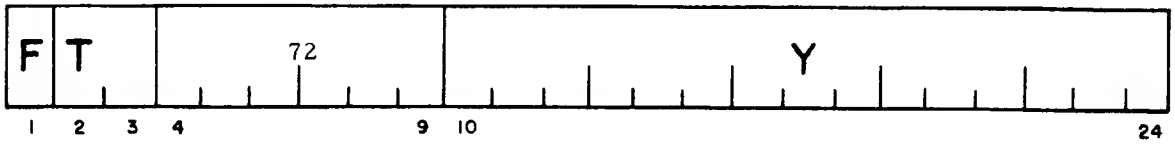

DESCRIPTION: The contents of the specified index register replace the address portion of the contents of Y, bits 10 through 24. The contents of the index register are unchanged.
TAG: The TAG portion of this instruction specifies which index register will be stored. If no TAG is specified (bits 2 and 3 are ZERO), the address portion of the contents of Y, bits 10 through 24, will be replaced with all ONEs (unless indirect addressing is specified, see below).
FLAG: If this instruction is flagged for indirect addressing, the normal procedure for forming effective address will be followed and the address portion of the contents of Y, bits 10 through 24, will be replaced by the contents of the index register specified in the TAG portion of the contents of Y, bits 2 and 3.
INDICATORS: None




DESCRIPTION: The contents of Y, bits 10 through 24, are incremented by ONE and the resulting sum, modulo $2^{15}$, replaces the contents of Y, bits 10 through 24. The contents of Y, bits 1 through 9, are unchanged. Thus, it is possible to have several index registers in memory that can be incremented, saved, and made available for use in indexing operations, all with one instruction.
TAG: If this instruction is tagged, the incremented contents of Y, bits 10 through 24, will replace the contents of the specified index register. If no TAG is specified (bits 2 and 3 are ZERO), the incremented value will replace the address portion of the A-register, bits 10 through 24, and the command portion of the A-register, bits 1 through 9, will be set to ZEROs.

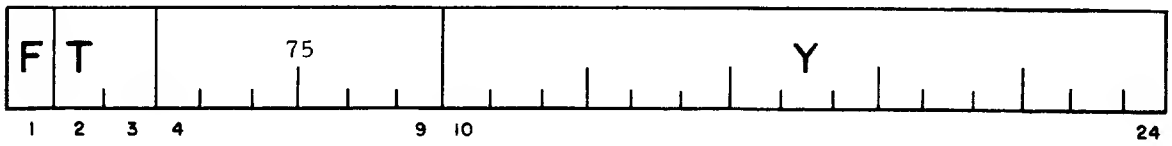
FLAG: If this instruction is flagged for indirect addressing, the normal procedure for effective address forming will be followed and the TAG portion of Y, bits 2 and 3, will be interpreted as discussed under TAG.
INDICATORS: None

JIX JUMP ON INDEX 2.10 μsec




DESCRIPTION: If the contents of the specified index register are not ZERO, the computer takes its next instruction from Y and continues from there. If the contents of the specified index register are ZERO, the computer takes the next sequential instruction.
TAG: The TAG specifies which index register is to be examined. If no TAG is specified (bits 2 and 3 are ZERO), this instruction is treated as an NOP (unless indirect addressing is specified, see below).
FLAG: If this instruction is flagged for indirect addressing, the normal procedure for effective address forming will be followed and the address portion of the contents of Y, bits 10 through 24, will specify the jump destination, if a jump occurs. The TAG portion of the contents of Y, bits 2 and 3, specifies which index register will be examined by this instruction.
INDICATORS: None

JXI JUMP ON INDEX INCREMENTED 2.80 μsec

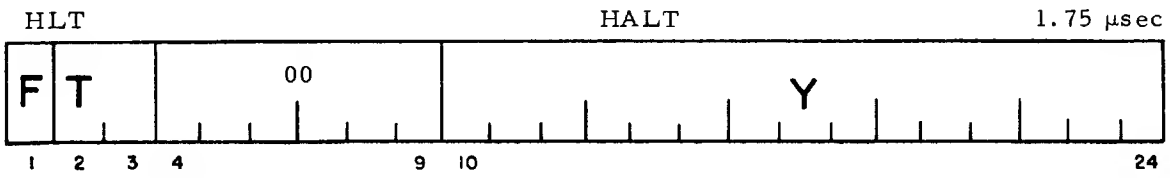



DESCRIPTION: The contents of the specified index register are incremented by ONE and the resulting sum replaces the contents of the specified index register. If, after incrementing, the contents of the specified index are not ZERO, the computer takes its next instruction from Y and continues from there. If the contents of the specified index register are ZERO (after incrementing), the computer takes the next sequential instruction.
TAG: The TAG specifies which index register is to be modified. If no TAG is specified (bits 2 and 3 are ZERO), this instruction is treated as an NOP (unless indirect addressing is specified, see below).
FLAG: If this instruction is flagged for indirect addressing, the normal procedure for effective address forming will be followed and the address portion of the contents of Y,

bits 10 through 24, will specify the jump destination, if a jump occurs. The TAG portion of the contents of Y, bits 2 and 3, specifies which index register will be affected by this instruction.

INDICATORS: None

## CONTROL INSTRUCTIONS




DESCRIPTION: The computer will halt until the START button on the operation console is depressed, at which time execution will be resumed at the next sequential instruction. The address portion of this instruction, bits 10 through 24, is not interpreted unless indirect addressing is specified.

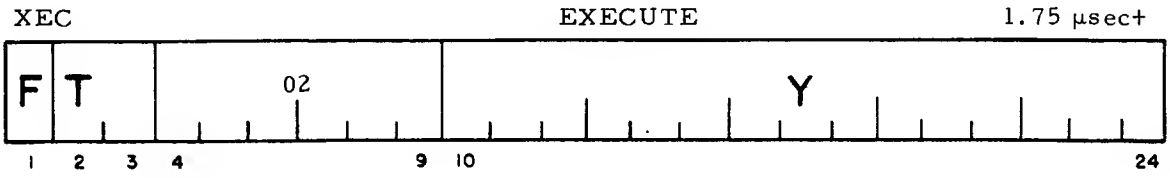
NOTE

If the computer is in an interrupt condition, this instruction will be treated as an NOP. If an interrupt occurs while the computer is in a halt condition, the interrupt will be honored. At the conclusion of the interrupt program, the computer will return to the HLT.

TAG: Does not affect the instruction.

FLAG: Does not affect the instruction, but causes the DDP-124 to execute a non-significant address modification loop.

INDICATORS: Stop indicator.




DESCRIPTION: The instruction at Y is executed. The computer then takes the next sequential instruction following the XEC instruction, and continues from there. If the instruction at Y results in a jump, the computer takes its next instruction from the jump destination and continues from there. If the instruction at Y is a skip instruction, the skip, if any, will be relative to the XEC instruction and not the instruction at Y. The contents of Y, bits 1 through 24, are not changed.

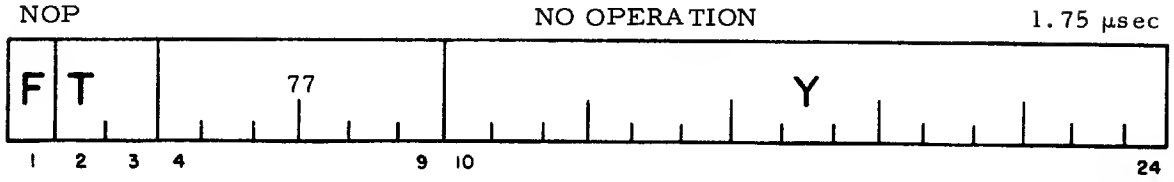
TAG: Normal

FLAG: Normal

INDICATORS: None

NOTE

W = Time of executed instruction.



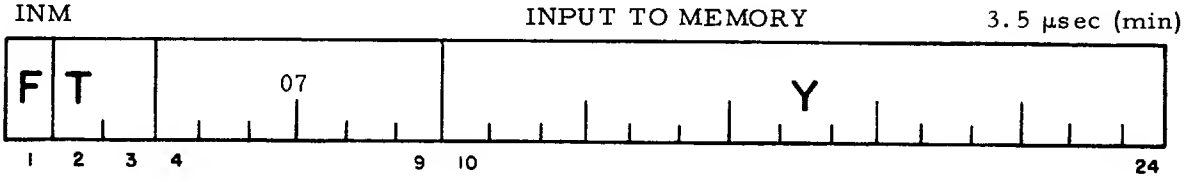

DESCRIPTION: No operation is performed by this instruction. The computer will take the next sequential instruction and continue from there. The address portion of this instruction, bits 10 through 24, are not interpreted unless indirect addressing is specified.

TAG: Does not affect the instruction.

FLAG: Does not affect the instruction, but delays fetching of the next sequential instruction.

INDICATORS: None

## INPUT/OUTPUT INSTRUCTIONS




DESCRIPTION: The contents of the buffer attached to the enabled input channel replace the contents of Y. If the buffer is a word buffer, the input data will replace the contents of Y, bits 1 through 24. If the buffer is a partial word buffer, the input data will replace the low-order contents of Y and set the remaining portion of Y to ZEROs. If the buffer is not ready, this instruction will wait until a READY signal is received from the buffer before completing its execution.

NOTE

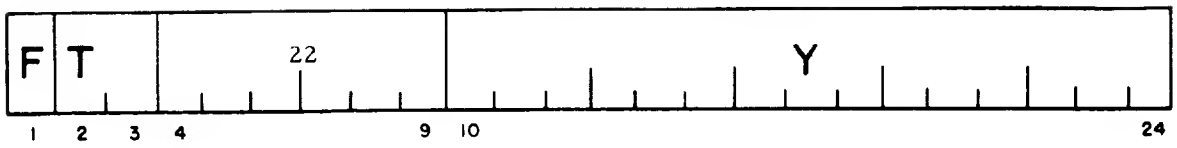
The selection of the device that will supply the input data and the enabling of the input channel connected to the device are accomplished by OCP instructions (see OCP).

TAG: Normal

FLAG: Normal

INDICATORS: Upon completion of this instruction, the READY indicator associated with the input channel will be reset.

OTM OUTPUT TO MEMORY 3.5 μsec (min)




DESCRIPTION: The contents of Y replace the contents of the buffer attached to the enabled output channel. If the buffer is a word buffer, the contents of Y, bits 1 through 24, will replace the contents of the buffer. If the buffer is a partial word buffer, the low-order contents of Y will replace the contents of the buffer. If the buffer is not ready, this instruction will wait until a READY signal is received from the buffer before completing its execution. The contents of Y, bits 1 through 24, are unchanged.

NOTE

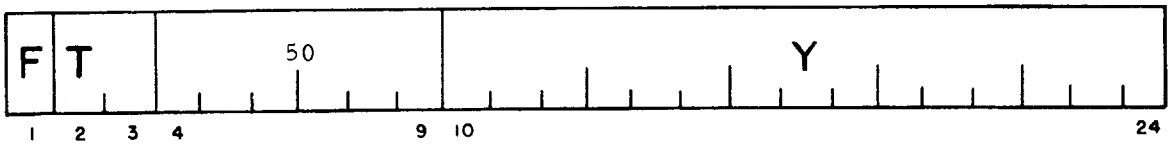
The selection of the device that will receive the output data and the enabling of the output channel connected to the device are accomplished by OCP instructions (see OCP).

TAG: Normal

FLAG: Normal

INDICATORS: Upon completion of this instruction, the READY indicator associated with the output channel will be reset.

OTA OUTPUT FROM A 2.10 μsec (min)




DESCRIPTION: If bit position 10 of this instruction contains a ZERO, Y will not be interpreted (unless indirect addressing is specified, see below) and the contents of the A-register replace the contents of the buffer attached to the enabled output channel. If the buffer is a word buffer, the contents of the A-register, bits 1 through 24, will replace the contents of the buffer. If the buffer is a partial word buffer, the low-order contents of the A-register will replace the contents of the buffer. If the buffer is not ready, this instruction will wait until a READY signal is received from the buffer before completing its execution. The contents of the A-register, bits 1 through 24, are unchanged.

If bit position 10 of this instruction contains a ONE, bits 19 through 24 are used to specify which of the 6 bits of information are to be transmitted from the low-order portion of the A-register, bits 19 through 24, to the low-order portion of the enabled buffer. For each bit position of this instruction, bits 19 through 24, that contains a ONE, the contents of the corresponding bit position of the A-register will be transmitted to the buffer. For each bit position of this instruction, bits 19 through 24, that contains a ZERO, a ZERO will

be transmitted to the corresponding bit position of the buffer. Any or all of the low-order 6 bits of information may be transmitted in this manner.

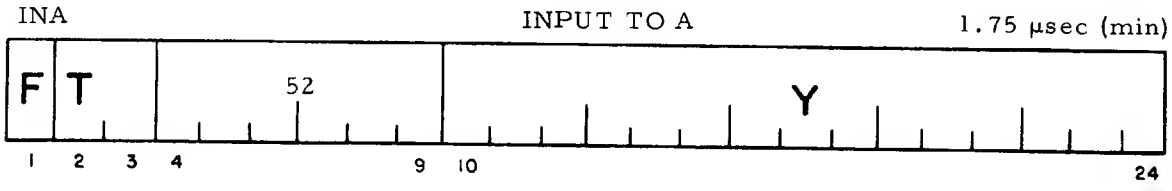
NOTE

The selection of the device that will receive the output data and the enabling of the output channel connected to the device are accomplished by OCP instructions (see OCP).

TAG: The TAG portion of this instruction is not interpreted unless indirect addressing is specified.

FLAG: If this instruction is flagged for indirect addressing, the normal procedure that is used to form an effective address will be followed and the address portion of the memory word at the effective address will be interpreted as described above for bit positions 10 and 19 through 24.

INDICATORS: Upon completion of this instruction, the READY indicator associated with the input channel will be reset.




DESCRIPTION: If bit position 10 of this instruction contains a ZERO, Y will not be interpreted (unless indirect addressing is specified, see below) and the contents of the buffer attached to the enabled input channel replace the contents of the A-register. If the buffer is a word buffer, the input data will replace the contents of the A-register, bits 1 through 24. If the buffer is a partial word buffer, the input data will replace the low-order contents of the A-register and set the remaining portion of the A-register to ZEROs. If the buffer is not ready, this instruction will wait until a READY signal is received from the buffer before completing its execution.

If bit position 10 of this instruction contains a ONE, bits 19 through 24 are to be used to specify which of the 6 bits of information are to be transmitted from the low-order portion of the enabled buffer to the low-order portion of the A-register, bits 19 through 24. For each bit position of this instruction, bits 19 through 24, that contains a ONE, the contents of the corresponding bit position of the buffer will replace the contents of only that position in the A-register (no other position in the A-register will be changed). Any or all of the low-order 6 bits of information may be transmitted in this manner.

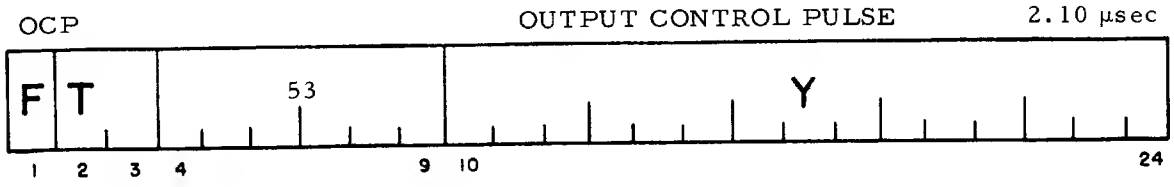
NOTE

The selection of the device that will supply the input data and the enabling of the input channel connected to the device are accomplished by OCP instructions (see OCP).

TAG: The tag portion of this instruction is not interpreted unless indirect addressing is specified.

FLAG: If this instruction is flagged for indirect addressing, the normal procedure that is used to form an effective address will be followed and the address portion of the memory word at the effective address will be interpreted as described above for bit positions 10 and 19 through 24.

INDICATORS: Upon completion of this instruction, the READY indicator associated with the input channel will be reset.

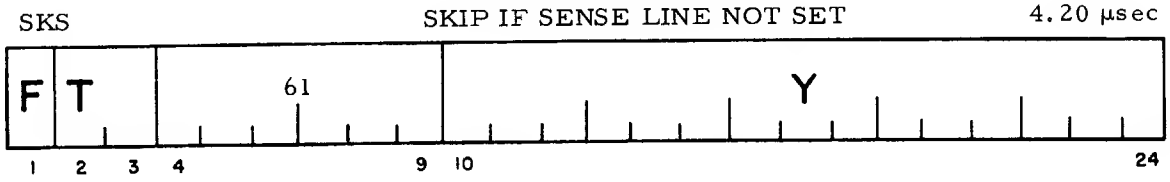



DESCRIPTION: An output pulse is generated by this instruction for the control of input/output channels and external equipment. The address portion, bits 10 through 24, specifies the unit to be selected, the type of control, etc. (See Appendix A for the OCP code assignments.)

TAG: Normal

FLAG: If this instruction is flagged for indirect addressing, the normal procedure that is used to form an effective address will be followed and the address portion of the memory word at the effective address will be interpreted as described above for bit positions 10 through 24.

INDICATORS: Certain indicators can be set with the OCP instruction (see Appendix A).




DESCRIPTION: The sense line specified by the address portion of this instruction, bits 11 through 24, is interrogated. If the sense line is not set, the computer skips the next instruction and continues from there; if the sense line is set, the computer will take the next sequential instruction. The lines that may be tested include 10 internal sense lines (6 sense switches, OVERFLOW indicator, IMPROPER DIVIDE indicator, input parity, and stop code), READY signals of input/output channels and external sense lines from peripheral equipment (busy status, parity errors, etc.). From 1 through 10 of the internal sense lines may be tested simultaneously (in group 0); if any or all of the lines are not set, the computer skips the next instruction. For group 1, similar simultaneous testing is also possible.

If the most significant address position of the instruction, bit 10, is a ONE, the flip-flop associated with the tested sense line is reset for certain assignments. (Appendix B contains the sense line selection assignments.)

TAG: Normal

FLAG: If this instruction is flagged for indirect addressing, the normal procedure that is used to form an effective address will be followed, and the address portion of the memory word at the effective address will be interpreted as described above for bit positions 10 and 11 through 24.

INDICATORS: Certain indicators can be reset with the SKS instruction (see Appendix B).

## INSTRUCTION TRAP FEATURE

The DDP-124 Instruction Trap Feature provides for program compatibility with the DDP-224. Octal operation codes for which no specific operation is implemented are treated as NOPs, except that, instead of proceeding to the next instruction, the DDP-124 stores the program counter and other processor status information in the "trap" location octal 10, and causes a jump to the transfer location octal 11. The contents of the trap location are as specified for an external priority interrupt trap location. (See pages 3-5, 3-6.)

# SECTION III
# INPUT/OUTPUT INFORMATION

## INPUT/OUTPUT CONTROL AND COMMUNICATION

The DDP-124 General-Purpose Computer establishes communication with peripheral (input/output) devices via sensing and control pulses. The sensing pulses (SKS codes) from the computer interrogate the input/output device to determine when the device is ready for operation. Control pulses (OCP codes) are then sent by the computer to start and stop the equipment and select operating modes and speeds. Communication may also be initiated by peripheral devices by means of interrupt lines (an optional feature).

When a communication link has been established, data transfer between the DDP-124 and the peripheral device can be accomplished by means of the following commands.

INA (input to A-register)
OTA (output from A-register)
INM (input to memory)
OTM (output from memory)

Should any of these commands occur when the selected data channel is not ready, execution of the command is delayed until the channel becomes ready. This delay can be avoided by sensing the status of the data channel with an SKS command before attempting to initiate communication.

An input/output character buffer, and an output bus are provided as standard equipment in every DDP-124. The character buffer transfers 6-bit characters between the computer and the selected peripheral device. The character buffer is connected to the typewriter, paper-tape punch, and paper-tape reader, and is used by these devices in on-line operations. Selection of one of these devices by an OCP automatically enables the character buffer. An interlock is provided so that selection of any of these three devices will deselect any other one. Additionally, an OCP code ('02070)* is available to disconnect all three devices.

The status of the character buffer can be tested with an SKS '14000; i.e., the sense line is set when the buffer is ready to transfer a word to or from the computer. The status of the connected character device can be tested with an SKS '2000.

Optional data channels are available to facilitate communication between the DDP-124 and peripheral devices; these are discussed in detail in this section under "Optional Data Channels."

*The apostrophe indicates an octal number.

## OCP Operations

As explained in Section II, the command word of the DDP-124 contains an OP-code portion and an address portion. The OP-code portion contains the coded designation of the operation that the computer is to perform. The address portion in most cases contains the address of the information (operand) with which the operation is to be performed. However, when the command word contains an OP-code for an output control pulse (octal 53), the address portion of the command word contains the code for the particular I/O device control line on which the output control pulse is to be sent.

When an OCP command is executed, a 0.70-μsec pulse is applied to the particular line specified in the command word. The OCP line is, in turn, connected to a control circuit on the external device which, upon receipt of the pulse, performs the desired action, such as starting a tape reel, start reading a disk file, stopping the movement of a feed mechanism, or closing and holding a switch to an I/O channel. For example, the following instruction should be executed to enable the typewriter for an input operation:

```
OCP '2000      ENABLE TYPEWRITER AND CHARACTER BUFFER
```

After the execution of this instruction, any key depressed on the typewriter will cause the code for the equivalent character to be transmitted to the character buffer channel. When the character buffer has received the code, the READY indicator associated with the character buffer will be set. This action informs the central processor that the character is available and is ready to be transmitted. An INA or INM instruction should then be executed which will "unload" the character buffer and reset the READY indicator. The character buffer can now accept another code from the typewriter. When an input instruction (INA or INM) is executed before a character is in the buffer (READY not set), the input instruction will wait for the ready signal before unloading the buffer. This is called an input/output interlock. All input/output instructions in the DDP-124 will interlock until a READY signal is received. Thus, although it may take milliseconds, or even seconds, between the selection of a device and the first character, the input instruction automatically waits until the character has been received.

NOTE

A complete list of OCP address codes is included in Appendix A of this Programmer's Manual.

## SKS Operations

Not only is it necessary to control the operation of the external devices so that information can be transferred into and out of the computer at the proper times, it is also necessary for the computer to sense the condition of the device so that the computer will not waste time while the external device is either preparing to operate or is doing an operation that does not occupy the full attention of the computer. The sensing of the state of external devices is possible through the use of sense lines connected to sense circuits at the external equipment and through the use of SKS commands. The sense lines are connected to the

external devices so that when the external device is busy, the sense line will carry a 0-volt level, and when it is not busy the sense line will carry a +6-volt level. For the computer to tell whether a device is busy or not, an SKS command must be executed. The SKS abbreviation means "skip if sense line not set," in this case, the "set" meaning that the device is busy (+6 volts on sense line).

Like an OCP command, an SKS command contains the coded SKS operation (61) in the OP-code bits portion of the command word and the designation of the sense line in the address portion of the command word. Sense lines provided as standard equipment on the DDP-124 are as follows.

1. Two lines for the standard I/O character channel.
2. One line for the standard character devices (typewriter, tape punch, tape reader).
3. Six lines for the sense switches on the control panel.
4. Four lines for internal flip-flops (overflow, improper divide, input parity, and stop code).

All or several of the sense lines in (3) and (4) above can be tested simultaneously. Any or all of the tested sense lines may cause a skip. For example, SKS 00001 tests the status of sense switch number one, and SKS 00002 tests the status of sense switch number two. Hence, SKS 00003 will cause a skip only when both sense switches one and two are in the OFF position.

Up to 64 sense lines can be added. Optional equipment which requires sense line inputs to the computer generally includes those sense lines with the option.

NOTE

A complete list of sense line codes is included in Appendix B of this Programmer's Manual.

## Peripheral Device Codes

Table 3-1 shows the various characters used by the several most common peripheral devices and the equivalent octal codes used by the DDP-124 to identify the characters. Columns 2 and 3 contain the characters for the typewriter in the lower and upper cases. A coded instruction must first select whether lower case or upper case is desired, and this is done by an octal code 74 for lower shift and 75 for upper shift, as shown. Column 4 indicates the Hollerith card code, one of the codes that can be used by the card reader or card punch to express the various characters. A comma, for example, is expressed in Hollerith code by a hole in the 0, 8, and 3 rows of the card. Column 7 of the table indicates the corresponding IBM compatible DDP-124 magnetic tape code.

Table 3-1.
Peripheral Device Codes

| 1 DDP Memory Octal Code | 2 Typewriter Lower Case | 3 Typewriter Upper Case | 4 Card Code (Hollerith) | 5 Symbol on Keypunch | 6 Line Printer | 7 Magnetic Tape Code (BCD) |
|---|---|---|---|---|---|---|
| 00 | 0 | b | 0 | 0 | 0 | 0* |
| 01 | 1 | | 1 | 1 | 1 | 1 |
| 02 | 2 | | 2 | 2 | 2 | 2 |
| 03 | 3 | | 3 | 3 | 3 | 3 |
| 04 | 4 | : | 4 | 4 | 4 | 4 |
| 05 | 5 | @ | 5 | 5 | 5 | 5 |
| 06 | 6 | √ | 6 | 6 | 6 | 6 |
| 07 | 7 | > | 7 | 7 | 7 | 7 |
| 10 | 8 | | 8 | 8 | 8 | 8 |
| 11 | 9 | | 9 | 9 | 9 | 9 |
| 12 | | | All Other Codes | ? | ? | Nonexistent* |
| 13 | # | - | 8-3 | # | = | = |
| 14 | | | 8-4 | @ | ' | ' |
| 15 | | | 8-5 | : | : | |
| 16 | | | 8-6 | > | ! | |
| 17 | | | 8-7 | ≥ | > | |
| 20 | * | ¢ | 11-8-4 | * | * | Blank |
| 21 | / | | 0-1 | / | / | / |
| 22 | S | | 0-2 | S | S | S |
| 23 | T | | 0-3 | T | T | T |
| 24 | U | = | 0-4 | U | U | U |
| 25 | V | % | 0-5 | V | V | V |
| 26 | W | " | 0-6 | W | W | W |
| 27 | X | ' | 0-7 | X | X | X |
| 30 | Y | | 0-8 | Y | Y | Y |
| 31 | Z | | 0-9 | Z | Z | Z |
| 32 | ' | | 0-8-2 | ≠ | " | |
| 33 | | | 0-8-3 | , | , | , |
| 34 | | | 0-8-4 | % | ( | ( |
| 35 | | | 0-8-5 | = | = | |
| 36 | TAB | | 0-8-6 | ] | ] | |
| 37 | | | 0-8-7 | " | " | |
| 40 | - | - | 11 | - | - | - |
| 41 | J | | 11-1 | J | J | J |
| 42 | K | | 11-2 | K | K | K |
| 43 | L | | 11-3 | L | L | L |
| 44 | M | ) | 11-4 | M | M | M |
| 45 | N | * | 11-5 | N | N | N |
| 46 | O | Δ | 11-6 | O | O | O |
| 47 | P | ; | 11-7 | P | P | P |
| 50 | Q | | 11-8 | Q | Q | Q |
| 51 | R | | 11-9 | R | R | R |
| 52 | tab | | 11-0 | | ; | o |
| 53 | $ | | 11-8-3 | $ | $ | $ |
| 54 | Backspace | | 11-8-6 | ;(Stop Code) | √ | * |
| 55 | | | 11-8-5 | ) | [ | |
| 56 | Space | | Blank | Blank | Blank | |
| 57 | | | 11-8-7 | ≤ | < | |
| 60 | & | & | 12 | & | + | + |

Table 3-1. (Cont)
Peripheral Device Codes

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| DDP Memory Octal Code | Typewriter | | Card Code (Hollerith) | Symbol on Keypunch | Line Printer | Magnetic Tape Code (BCD) |
| | Lower Case | Upper Case | | | | |
| 61 | A | | 12-1 | A | A | A |
| 62 | B | | 12-1 | B | B | B |
| 63 | C | | 12-3 | C | C | C |
| 64 | D | ( | 12-4 | D | D | D |
| 65 | E | □ | 12-5 | E | E | E |
| 66 | F | ‡ | 12-6 | F | F | F |
| 67 | G | < | 12-7 | G | G | G |
| 70 | H | | 12-8 | H | H | H |
| 71 | I | | 12-9 | I | I | I |
| 72 | | | 12-0 | + | ↑ | ŧ |
| 73 | | | 12-8-3 | . | . | . |
| 74 | Lower Shift | | 12-8-4 | [ | ) | ) |
| 75 | Upper Shift | | 12-8-5 | ( | % | |
| 76 | Car. Return | | 12-8-6 | < | \ | / |
| 77 | Line Feed | | 12-8-7 | ← | ← | |

*A zero recorded in BCD mode on magnetic tape will be translated into an octal 12 when written and back to zero when read. An octal 12 may be recorded in BCD mode on magnetic tape, but will be read back as zero.

## OPTIONAL DATA CHANNELS

A wide variety of optional input/output data channels are available to facilitate both synchronous and asynchronous transfer of information between the DDP-124 and peripheral devices. Some typical applications of these channels are illustrated by means of input/output programming examples in Section IV.

### Input/Output Character Buffers

A number of character buffer registers can be added to the DDP-124. These 6-bit buffers check and assign parity and operate in the ready mode or with interrupt lines. The character buffer can be implemented to handle 8-bit characters.

### Parallel Input/Output Channels

A large number and variety of parallel 24-bit input and/or output channels are available as options. The options include the following characteristics.

1. Parallel input or output channels operable in the ready or interrupt modes
2. Parallel input or output buffer register and channels operable in the ready or interrupt mode

## Direct Memory Access Channels

Direct memory access (DMA) channels can be added to transfer input/output data directly into or out of memory with no need for program control. The input/output control unit of the DMA includes a parallel, 24-bit information buffer register which transfers data to or from the memory information register (MIR); an address selection register which provides address information to the memory address register (MAR); a limit register for controlling the number of words transferred; and a flip-flop for input or output mode control.

When the DMA requests memory access, the request will be honored immediately after the memory completes its current read or write function. The next fetch or execute cycle is delayed for the memory cycle required to read or write the DMA word. Hence, the extent of interference with the computer program is determined by the data transfer rate of the I/O device. If, for example, a line printer is operated through a DMA channel, the computer can process several thousand instructions between DMA requests for memory access to read each 30-word data block. The DMA can transfer data at rates up to 525,000 24-bit words per second, and may operate in either input or output mode.

Communication through a DMA channel is established as follows. The busy status of the DMA is sensed by an SKS command. When the DMA is not busy, an OCP command may be issued to enable DMA. Next, an OTM or OTA command loads the DMA address register with a number which specifies the starting address of the data block to be transferred. The DMA limit register is then loaded by a second OTM or OTA command with a number which specifies the number of words to be transferred. The I/O device connected to the DMA channel may then be started with an OCP command, and the data transfer is processed automatically while the program continues with other functions.

If bit 2 of the word transferred by the first OTM or OTA command is a ONE, the limit register will be loaded with all one's, and a second OTM or OTA is not necessary. If bit 1 of the first word transferred in a ONE, the DMA will be conditioned for the output mode; if bit 1 is a ZERO, the DMA will be conditioned for the input mode. If bit 3 of the first word transferred is a ONE, the DMA will operate in the HOG mode. The HOG mode gives the DMA sole access to memory, enabling the DMA to operate at its maximum data transfer rate.

## Word-Forming Buffer

Word-forming buffers (WFB) for both input and output are available. Tne buffer automatically forms a computer word from input characters or forms output characters from a computer word (depending upon the direction of transfer), checks parity and generates output characters with parity from computer words. The WFB-selection command (an OCP) specifies either 1, 2, 3, or 4 6-bit characters per word to be transferred. The control also provides a buffer-ready signal and the checking and generation of character parity. The word-forming buffer is capable of buffering the flow of 6-bit words between a peripheral device and the computer, or acting as an interface between the character-sequential operation of peripheral devices, and the word-sequential operation of the

computer. Special word-forming buffers are available for operating on 8-bit and 12-bit characters (see Appendix A).

## OPTIONAL INTERRUPT FACILITIES

One or two eight-line interrupt option modules may be added to the DDP-124, permitting peripheral devices or external events to signal the program. When an external event activates an interrupt line, the computer stores the contents of the program counter and other processor status information at a "trap" location, and transfers control to another location. Both the trap and transfer locations are uniquely associated with the interrupt line. For example, the trap and transfer locations for interrupt lines 0 through 3 are:

| Interrupt Line | Trap Location | Transfer Location |
|---|---|---|
| 0 | 00020 | 00021 |
| 1 | 00022 | 00023 |
| 2 | 00024 | 00025 |
| 3 | 00026 | 00027 |

The contents of a trap location are as follows:

Bits 1 through 3 — The values for the three flip-flops associated with the SMP instruction, at the time of interrupt:

- Bit 1: ZERO indicates SUB; ONE indicates ADD.
- Bit 2: ZERO indicates complement of magnitude may be necessary; ONE indicates complement of sign is necessary.
- Bit 3: ZERO indicates a carry occurred; ONE indicates no carry occurred.

Bits 4 through 6, 7 through 9, 10 through 24 — The contents of the program register at the time the interrupt was honored are stored in this field.

The transfer location normally contains a JMP to the program which services the interrupting device.

The interrupt program must be terminated by a JRT command, containing the trap location in its address field, to ensure that the data stored at the trap location is used to restore the condition of the associated controls. The interrupt subroutine should generally provide for saving the contents of the A-, B-, and index registers.

Priority Assignments. Priority assignments in the 8-line system are sequential, with line 0 having the highest priority. If two lines are activated simultaneously, the line with the highest priority will be serviced first. Interrupt lines of higher priority will interrupt lines of lower priority.

Interrupt Enable. Interrupts are enabled and disabled under control of the interrupt switch on the console, or by program execution of enabling OCPs and disabling SKSs.

Selective Inhibit. OCP codes are available in pairs for inhibiting single interrupt lines, and for removing the inhibits. Thus, an OCP '2540 will cause the computer to ignore interrupt signals on line 0. Subsequently, an OCP '2541 will remove the inhibit, and the computer will once again respond to interrupt signals on line 0.

## TYPEWRITER

One 15-character-per-second revolving ball typewriter (Figure 3-1) is available as an option with each system for communicating with the computer by means of the input/output character buffer channel. This typewriter is a compact, light weight binary-coded-decimal (BCD) machine that utilizes a type head and keyboard arrangement designed primarily for use in conjunction with a computer. It has a 15-in. carriage with a 13-in. writing line; the ribbon mechanism has three ribbon lift positions and a stencil position. Both ribbon and spool are contained in a snap-in cartridge to allow rapid replacement.

The selection mechanism is used during input and output operations. When the typewriter is used as an output device, the mechanism selects the character to be printed. During input operations, which are initiated manually, contacts in the selection mechanism transmit pulses that define the character being printed. During output operations the computer transmits pulses to the typewriter to be selected and printed.

Figure 3-1. Input/Output Typewriter

### Character Transfer

Fourteen data lines are used to transfer data between the typewriter and the computer. Seven lines carry the 6-bit character codes plus parity, and seven lines carry the following machine function signals: tab, space, carriage return, index, upper case shift, lower case shift, and backspace. The character channel incorporates logic to encode machine function signals as 6-bit characters.

## Sensing and Control

Output control pulses and sense lines control and monitor the operation of the typewriter. The following OCP signals are used:

OCP '02000: Typewriter input select (keyboard and character buffer enabled)

OCP '02010: Typewriter output select (keyboard inhibited and character buffer enabled)

OCP '02070: Disconnect all three standard devices on standard input/output character buffer channel

NOTE

Selection of a standard device on the standard input/output character buffer channel automatically deselects any other standard device which is connected to it.

Two SKS signals are used with the typewriter: an SKS STOP CODE and an SKS BUSY. The SKS STOP CODE line (SKS '01000) is turned on whenever the backspace key on the typewriter is depressed in the input mode. The SKS BUSY signal (SKS '02000) is produced at the following times during the input transfer and control cycles (i.e., input to computer):

1. From beginning of a cycle (a cycle being one operation of the mechanism initiated by a key depression or output command from the computer) to the time at which the character buffer ready signal becomes true (about 30 ms).
2. For a duration of 6 ms immediately following deselection of the typewriter from the input mode.

Similarly, the SKS BUSY signal is produced at the following times during the output cycles:

1. From the instant that the character buffer is loaded until the typewriter no longer requires the contents of the character buffer (about 20 ms).
2. From the beginning of the cycle until the typewriter mechanism stops moving and can accept a new character.

NOTE

SKS '02000 is also present when either the paper-tape reader or punch is busy.

## Keyboard Lock

When the typewriter is on-line, the keyboard is locked, except when the typewriter is selected for input. When the typewriter is off-line with paper-tape reader, the keyboard is locked. If the typewriter is off-line alone or with the paper-tape punch, the keyboard is unlocked.

## PAPER-TAPE READER

A unidirectional perforated-tape reader (Figure 3-2) is standard in the DDP-124. The unit reads eight data channels (including parity), plus a sprocket hole channel, at the rate of 30 in./sec (10 characters per inch, or 300 characters per second). In the pulsed mode, the reader stops after reading every character and, consequently, reads at a much lower average speed.

The unit uses standard paper or mylar tapes (black paper recommended) 0.004 in. to 0.005 in. thick. The tape can be loaded without removing power by rotating a front mounted READY-LOAD switch to the LOAD position.

### Reader Modes

The paper-tape reader can operate in either of two on-line modes: a continuous mode or a pulsed mode. In the continuous mode the reader is started by the computer and runs at 300 characters per second. In the pulsed mode, the reader starts and stops for the reading of each character.

In the continuous mode, the reader will start reading at the following times.

1. After an OCP START READER
2. After the READER START button is pressed
3. After the FILL button is pressed

In the continuous mode the reader will stop reading at the following times.

1. After an OCP DISCONNECT is received (disconnects all peripheral devices on the character buffer channel)
2. After an OCP select for any other device on the character buffer channel
3. After a "stop code" is detected on paper tape
4. When the MASTER CLEAR button is depressed

If the reader receives a stop signal under any of the first three conditions within 150 μsec after reading a character, the tape will stop with the next character ready to be read. Otherwise (longer than 150 μsec), the next character will be skipped before stopping.

In the pulsed mode the reader is automatically stopped by associated control circuits every time a character is read. It restarts upon the receipt of an OCP START READER or after the START button is pressed.

### Sensing and Control

The paper-tape reader starts reading on receipt of an OCP '2100. A sense line is coupled to the stop code output line of the reader and is interrogated by the computer with SKS '1000. The line will be set after a stop code is read (a hole in channel eight of the tape). When the paper tape is in motion, the character buffer device busy sense line is set and can be interrogated by SKS '2000. Character pulse codes are shown in Table 3-2.


Figure 3-2. Paper-Tape Reader

Table 3-2.
Typewriter and Paper Tape Codes

| OCTAL CODE | TYPEWRITER L/C | TYPEWRITER U/C | PAPER TAPE 8 | 7 | 6 | P | 4 | S | 3 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | Ø | b | | | | o | | • | | | | |
| 01 | 1 | ■ | | | | | | • | | | o | |
| 02 | 2 | ■ | | | | | | • | | o | | |
| 03 | 3 | ■ | | | | o | | • | | o | o | |
| 04 | 4 | : | | | | | | • | o | | | |
| 05 | 5 | @ | | | | o | | • | o | | o | |
| 06 | 6 | √ | | | | o | | • | o | o | | |
| 07 | 7 | > | | | | | | • | o | o | o | |
| 10 | 8 | ■ | | | | | o | • | | | | |
| 11 | 9 | ■ | | | | o | o | • | | | o | |
| 13 | # | - | | | | | o | • | | o | o | |
| 20 | :: | ¢ | | | o | | | • | | | | |
| 21 | / | ■ | | | o | o | | • | | | o | |
| 22 | S | ■ | | | o | o | | • | | o | | |
| 23 | T | ■ | | | o | | | • | | o | o | |
| 24 | U | = | | | o | o | | • | o | | | |
| 25 | V | % | | | o | | | • | o | | o | |
| 26 | W | 11 | | | o | | | • | o | o | | |
| 27 | X | ' | | | o | o | | • | o | o | o | |
| 30 | Y | ■ | | | o | o | o | • | | | | |
| 31 | Z | ■ | | | o | | o | • | | | o | |
| 33 | , | ■ | | | o | o | o | • | | o | o | |
| 36 | tab | | | | o | o | o | • | o | o | | |
| 40 | - | - | | o | | | | • | | | | |
| 41 | J | ■ | | o | | o | | • | | | o | |
| 42 | K | ■ | | o | | o | | • | | o | | |
| 43 | L | ■ | | o | | | | • | | o | o | |
| 44 | M | ) | | o | | o | | • | o | | | |
| 45 | N | ¤ | | o | | | | • | o | | o | |
| 46 | O | Δ | | o | | | | • | o | o | | |
| 47 | P | ; | | o | | o | | • | o | o | o | |
| 50 | Q | ■ | | o | | o | o | • | | | | |
| 51 | R | ■ | | o | | | o | • | | | o | |
| 53 | $ | ■ | | o | | o | o | • | | o | o | |
| 54 | backspace | | | o | | | o | • | o | | | (SEE NOTE 1) |
| 56 | space | | | o | | o | o | • | o | o | | |
| 60 | & | & | | o | o | o | | • | | | | |
| 61 | A | ■ | | o | o | | | • | | | o | |
| 62 | B | ■ | | o | o | | | • | | o | | |
| 63 | C | ■ | | o | o | o | | • | | o | o | |
| 64 | D | ( | | o | o | | | • | o | | | |
| 65 | E | ¤ | | o | o | o | | • | o | | o | |
| 66 | F | ‡ | | o | o | o | | • | o | o | | |
| 67 | G | < | | o | o | | | • | o | o | o | |
| 70 | H | ■ | | o | o | | o | • | | | | |
| 71 | I | ■ | | o | o | o | o | • | | | o | |
| 73 | . | ~ | | o | o | | o | • | | o | o | |
| 74 | lower shift | | | o | o | o | o | • | o | | | |
| 75 | upper shift | | | o | o | | o | • | o | | o | |
| 76 | car. return | | | o | o | | o | • | o | o | | |
| delete | index | | o | o | o | o | o | • | o | o | o | (SEE NOTE 2) |
| stop | | | o | | | | | • | | | | (SEE NOTE 1) |

Note 1: In the output mode, the backspace code will cause the typewriter to backspace one character. In the input mode, depressing the backspace key sets the stop code flip-flop and causes a zero character to be transferred to the computer.

Note 2: In the output mode, the index code ($77_8$) will cause the typewriter to advance carriage one line. In the input mode, depressing the index key causes an index code ($77_8$) to be transferred to the computer.

## Paper-Tape Reader Specifications

The paper-tape reader has the following characteristics.

1. Reads at speeds up to 300 CPS
2. Asynchronous stepping at speeds to 60 CPS
3. Self-cleaning photo diodes
4. Variable 3-position detent action 0.687, 0.875, and 1.000 in. Reads 5, 6, 7, or 8 levels
5. Can stop on a character for all speeds
6. Start time, 5 ms
7. Temperature, 40° to 120°F
8. Reads paper or mylar tape 0.004 in. to 0.005 in. thick. Can be adjusted to read tapes from 0.0025 to 0.008 in.
9. Density of 10 characters per inch

## PAPER-TAPE PUNCH

A paper-tape punch is standard in the DDP-124 (Figure 3-3). The unit perforates 8-channel paper tape at a rate of 110 characters per second.

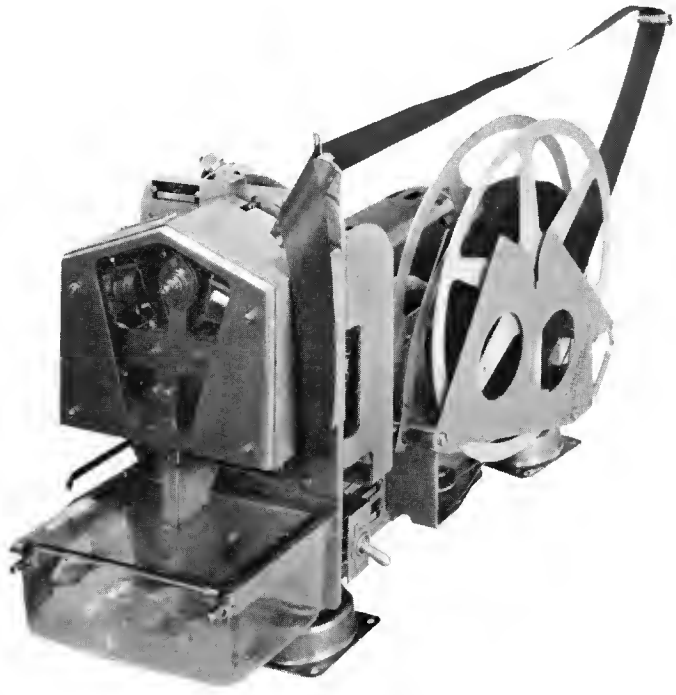


Figure 3-3. Paper-Tape Punch

### Sensing and Control

The paper-tape punch requires a single OCP command (OCP '2200) to start the punch mechanism and enable the character buffer. During punching and tape moving operations, the character device sense line is set, and can be tested by an SKS '2000.

Character pulses are applied to the punch from the computer in the coded forms of Table 3-2. For example, a character J would be received at the paper-tape punch as an octal 41, thereby activating the punches in tracks 1 and 7 along with the parity bit. A parity bit is punched whenever it is needed to make the total number of holes come out to an odd number.

Depressing the tape feed switch at the top of the punch will run the tape through the punch but no punching (except of sprocket holes) will take place. Special OCP codes are available for turning off punch power (OCP '02210), and feeding the tape one sprocket hole (OCP '02220).

### Paper-Tape Punch Specifications

The paper-tape punch has the following characteristics.

1. Punching is synchronous.
2. A 6-inch take-up reel is provided with the unit.
3. The perforator is enclosed in a buffered oil case.
4. Punch paper, foil, or mylar tape
5. Supply 1000 ft
6. Tape width, 0.687 in., 0.875 in., and 1.000 in.
7. Temperature, 40° to 120°F
8. Density of 10 characters per inch

## MAGNETIC TAPE SYSTEMS

Two types of magnetic tape transports (MTT) are available as options with the DDP-124: a 45 ips unit (shown in Figure 3-4) and a 75 ips unit (shown in Figure 3-5). Both types have two density modes of 200 and 556 bits per inch.

The Tape Control Unit (TCU) enables the computer to control the tape transport mechanism and to read or write on the tape. The TCU works with either the 45 ips or 75 ips tape transport through selective extractions or insertions of plug-in modules. The TCU contains an integral word-forming buffer. Prior to any tape control or read/write operations, the TCU word-forming buffer must be enabled for transfer of either one, two, three, or four characters per word. Data may be transferred between the TCU and the computer through the I/O transfer bus, or through an optional DMA channel.

At the 200 bits-per-inch setting it takes 111 μsec for the 45 ips MTT to read or write one "byte" of 6 bits on tape. Since the computer can operate at transfer speeds well above this level, transfer operations of the computer can be calculated from this rate. For instance, if an MTT is working in the four characters-per-word mode, the average time between the input to memory (INM) or output from memory (OTM) operations would be 4 x 111 μsec or 444 μsec. In the high-density level of 556 bits per inch it takes only 40 μsec to read or write one character on the tape instead of 111 μsec. Computer operation times can be calculated at the 556-bpi level by using the 40 μsec factor.

Table 3-3.
45 IPS MTT Specifications

**Tape**
- Width: 1/2 in.
- Reels: 10-1/2 in.
- File protect ring
- IBM compatible

**Tape Speed**
- Forward: 45 in./sec
- Reverse: 45 in./sec
- Rewind: Less than 3 min for complete reel

**Recording Density**
- Dual density
- High: 556 character per inch
- Low: 200 characters per inch

**Head**
- 7 tracks
- Separated read and write elements in same head
- Gives possibility to read while write (check back)
- Distance between read and write element = 0.3 in.

**Recording Format**
- IBM compatible
- NRZ1: Non-return to zero, change on one
- 6 data tracks
- 1 track for parity bits
- Inter-record gap = 3/4 in.
- Loadpoint = beginning of tape: reflective spot (photo sense element)
- End of tape = end of tape: reflective spot (photo sense element)

NOTE

The tape never stops with the read/write head in the middle of a record. At the completion of any tape moving operation, a built-in gap detection circuit stops the tape with the head properly spaced in the gap.

**Modes**
- Manual and automatic
  - Automatic: controlled by computer
  - Manual: Under pushbutton control
    - Rewind button
    - Reverse button
    - Loadpoint button

**Size of Cabinet**
- Height: 72 in.
- Depth: 25 in.
- Width: 26 in.

Table 3-4.
75 IPS MTT Specifications

| |
|---|
| The specifications on this MTT are identical to those of the 45 ips MTT (listed above) except for the following differences. |
| Tape Speed<br>Forward: 75 ips<br>Reverse: 75 ips<br>Rewind: |
| Modes<br>Remote: Controlled by computer control signals as with 45 ips MTT<br>Manual: Under following pushbutton control<br>REWIND LOAD<br>REV<br>STOP<br>FWD<br>Hi/Lo: Controls recording density of MTU |
| Size of Cabinet<br>Height: 64 in.<br>Depth: 26 in.<br>Width: 28 in. |
| Power Requirements<br>115 volts ±10% single phase; 60 or 50 cps ±1%<br>3 KW with compressor; 1.2 KW without compressor |

When using an MTT at 75 ips, the low-density read or write factor is 66.6 μsec and the high-density factor is 24 μsec. Average times between computer operations can be determined as described in the preceding paragraph.

## Tape Format

Figure 3-6 shows how data is organized into records and files on the tape. Six data tracks and one parity track are used with the spacing as shown.

## MTT Parity Checking

Two types of parity checking are used; longitudinal and lateral. When using either binary or BCD code, the bit of the longitudinal parity code will be ZERO, if there is an even number of bits in the record for that channel; and ONE, if there is an odd number of bits in the record for that channel. In lateral parity, when in the binary mode, the parity bit will be a ZERO for an odd number of ONE's in the byte and a ONE for an even number of ONE's in the byte. For lateral parities used in BCD codes, the parity bit will be a ONE for an odd number of ONE's in the byte, and a ZERO for an even number of ONE's in the byte.

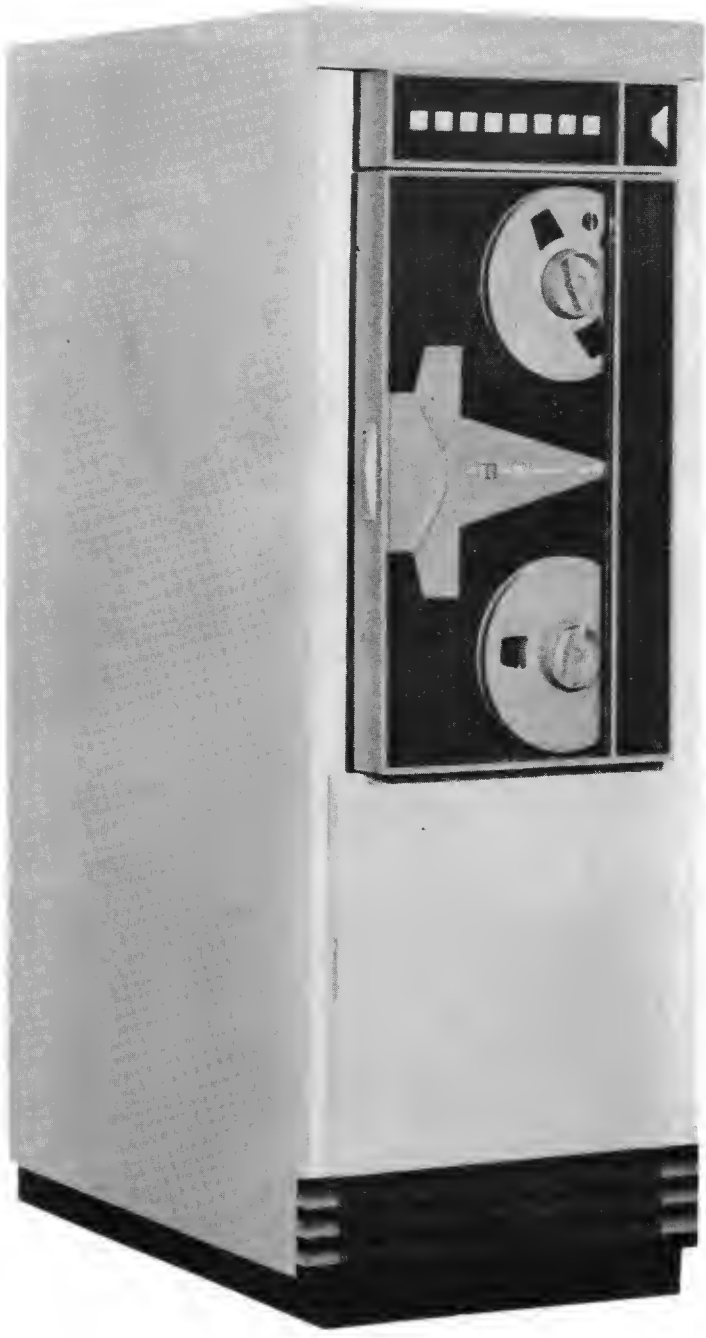


Figure 3-4. 45 IPS Magnetic Tape Transport (Single Unit)

Figure 3-5. 75 IPS Magnetic Tape Transport

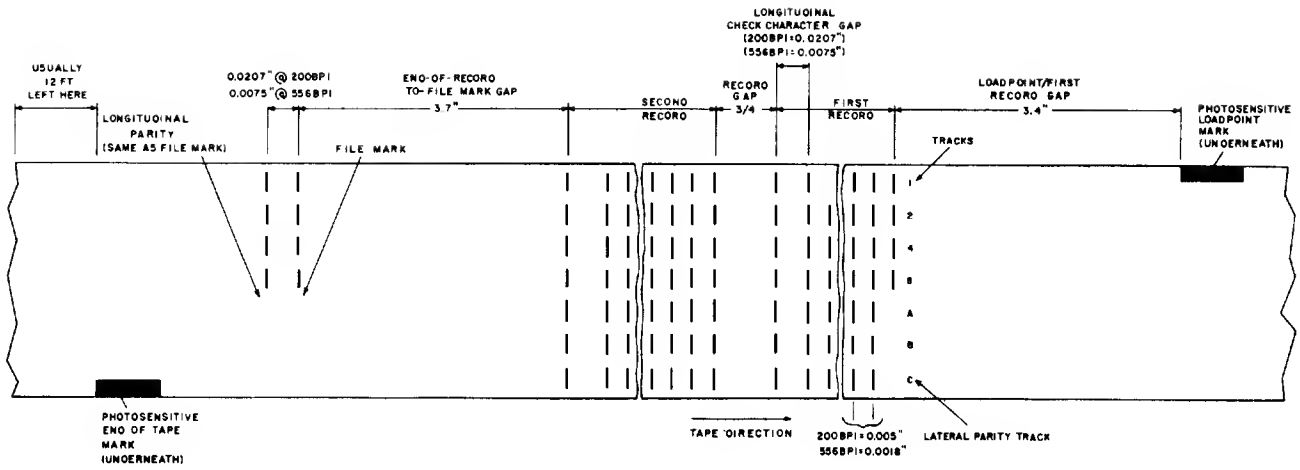



Figure 3-6. Magnetic Tape Format

Control Instructions for Magnetic Tape Transports

The computer uses 10 different output control pulse (OCP) codes to control the operation of magnetic tape transports. These are described in the following paragraphs.

Read One Record Binary. This output control pulse sets the tape handler in the forward drive mode. The characters are read and then transferred to the TCU character buffer. INM, INA, or FMB instructions may be used for inputting information.

The longitudinal parity check character at the end of a record is read by the MTT reading circuits, and sets a flip-flop in the TCU. When using a WFB and a file mark is read, it is transferred to the TCU, where it also sets a flip-flop. Both flip-flops may be tested by SKS commands.

Read One Record BCD. The BCD code is IBM compatible.

Forward One Record. This command allows the tape to be moved forward one record without tying up the channel. No information transfer takes place between tape and computer, only the forward movement of the tape to the next record gap. There is no longitudinal parity check.

Forward One File. This command allows the tape to be moved forward to the next file mark at which time the file mark is detected and the tape stops in the gap between the file mark and the next record. (See Figure 3-6.) As in the Forward One Record command, no information transfer takes place, and the associated computer is not tied up.

Rewind to Loadpoint. This command rewinds the tape to loadpoint. The MTT remains in the automatic mode after it reaches loadpoint so that it is ready for operation at the conclusion of the rewind. The channel is not busy when a tape is rewinding so that other

tapes on the channel may be used during a rewind. An SKS code has been assigned for each of the tape units for testing the rewind status of each unit.

Write One Record Binary. This instruction puts the MTT in the forward mode and the unit starts writing after a small delay that automatically constructs the proper size gap (3/4 in. from previous record or 3.4 in. from loadpoint to start of record). Prior to writing on the tape, the channel must be loaded by an OTA or OTM command. During the write operation, a read circuit allows for longitudinal parity checking of the record just written. A time delay is built in so that the head is about in the middle of the inter-record gap when the tape stops.

Certain operational restrictions are applicable to the write commands and must be taken into consideration by the programmer.

1. It is not permissible to write one record when the previous OCP command was a Backspace File. In such a case the programmer should first program Backspace One Record and then Forward One Record. He may then write one record.

2. It is not permissible to write one record when the previous OCP command was a Backspace One Record and that record happens to be a file mark. The programmer should first SKS the file mark. If the record was a file mark, he should then Backspace One Record, Forward One Record, and then write the record.

Write One Record BCD. Except for the code, this instruction is the same as Write One Record Binary. The BCD code is as described under Read One Record BCD.

Write File Mark. This instruction writes the file mark 0001111 (parity always ZERO) with the required gap in front of it. No output command is necessary; the file mark does not have to be stored in memory. The corresponding longitudinal parity character that follows the file mark is also an octal 17. (See Figure 3-6.)

Backspace One Record. This command causes the tape to reverse to the next previous record gap. No reading takes place; therefore, the channel is not busy during this operation. If a loadpoint is encountered before a record, tape motion stops. If the tape is at loadpoint, no motion takes place.

Backspace File. Same as Backspace One Record, except that tape stops at next previous file mark gap (detects file mark then stops). If a loadpoint is encountered before a file mark, tape motion stops. After backspacing a file, the tape is positioned so that the next forward instruction will encounter the file mark.

## Magnetic Tape Sense Signals

Six sense signals (SKS signals on DDP-124) are used with the magnetic tape transports as shown below.

1. SKS BUSY
2. SKS REWIND
3. SKS END OF TAPE
4. SKS LOADPOINT
5. SKS END OF FILE
6. SKS PARITY ERROR

SKS Parity Error. During writing, the tape is read back and longitudinal parity is checked. During reading, both lateral and longitudinal parity is checked. Whenever a parity error occurs in either parity check, a parity flip-flop is set, and the SKS parity signal can sense this. During writing, an SKS parity check should only be made when the tape unit is not busy. First an SKS busy check is made, and if the tape is not busy, SKS parity is checked. Lateral parity can be checked continuously during and after reading; longitudinal parity can only be checked after the end of an SKS BUSY signal.

SKS Busy. A busy signal indicates to the computer that the tape is moving. Five milliseconds after the tape stops, the SKS NOT BUSY signal is set. This means that it takes 5 ms after the tape stops before the tape can be started again by another OCP command.

SKS End of Tape. When the photosensitive End of Tape Mark is detected, an SKS end of tape flip-flop is set. This flip-flop is only reset by a rewind signal.

SKS Loadpoint. The signal on the SKS loadpoint sense line is set as long as the tape head is at the loadpoint mark on the tape.

SKS File Mark. When an end-of-file mark is detected the tape movement is stopped and the end of file flip-flop is set. This flip-flop is connected to the SKS file mark line. The flip-flop is reset on the beginning of the next tape movement.

SKS Rewind. Every individual tape transport of a system has its own rewind sense line. The line is set as long as the associated tape transport is rewinding tape; that is, until the loadpoint is reached.

## CARD READER

Two types of card readers are available with the DDP-124, depending upon user requirements for versatility and speed. One reader is 100 cpm unit (Figure 3-8) capable of reading 100 cards per minute. The other is a unit (Figure 3-7) capable of reading 200 cards per minute. Feed hoppers for both units have a capacity of 500 cards.

The Hollerith code divides the card up into 80 columns of 12 bits per column, each column containing the code for one character. When reading Hollerith-coded cards, the unit reads the 12-bit characters one by one and feeds the code through a built-in alphanumeric

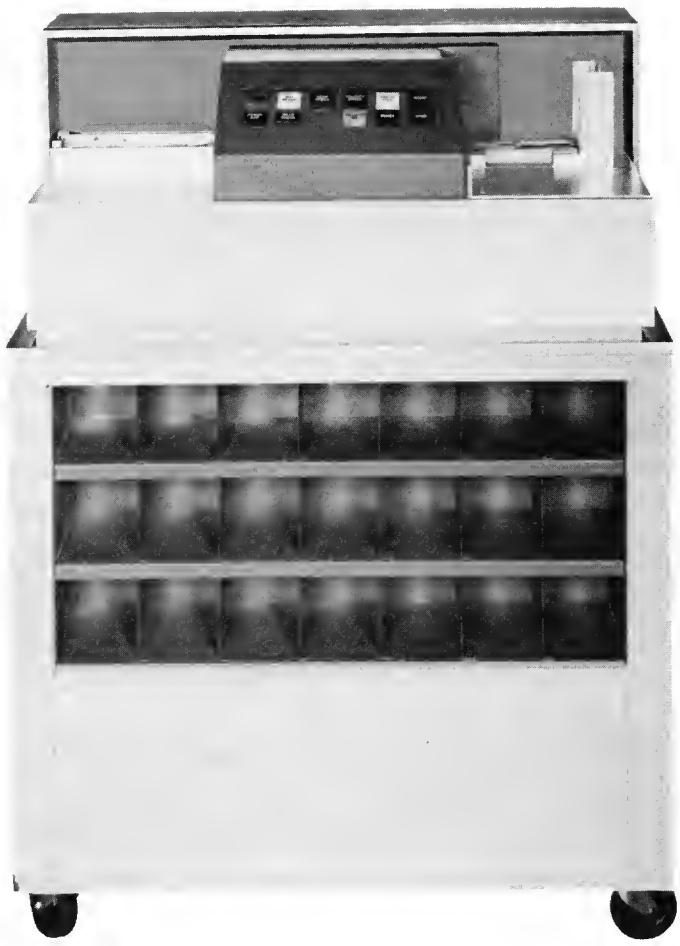


Figure 3-7. 200 CPM Punched Card Reader

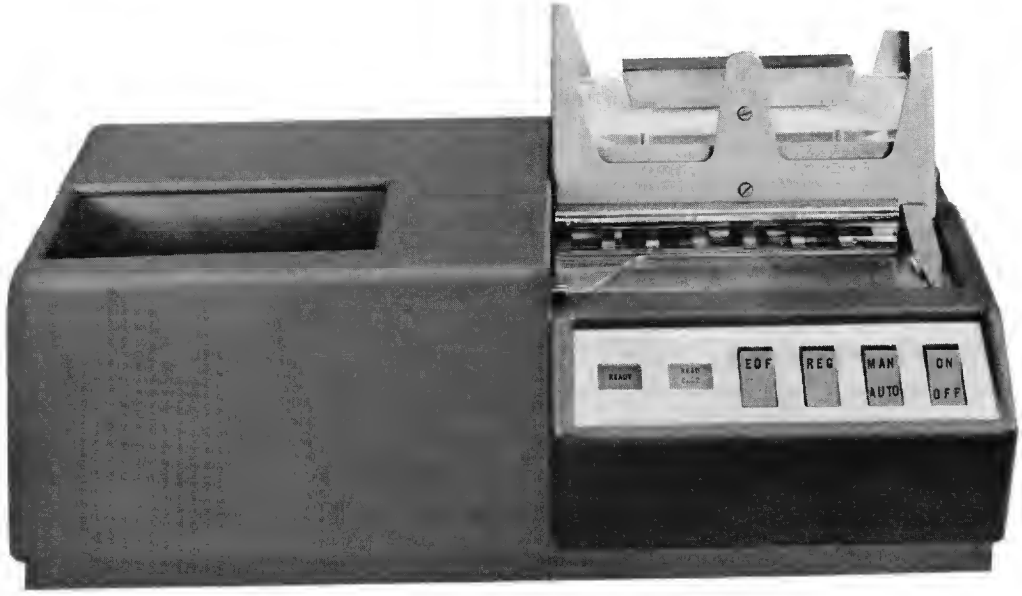



Figure 3-8. 100 CPM Card Reader

decoder. The decoder, in turn, produces a 6-bit coded character. Since this code is not compatible with the DDP-124, another decoding operation is necessary. This decoding takes place in the DDP-124 interface and converts the 6-bit code to a DDP-124 compatible 6-bit code.

In the binary system of encoding cards, each of the 80 card columns contains two 6-bit characters. The reader in this system strobes each column twice: first it extracts the uppermost six bits of the first column (rows 12, 11, 0, 1, 2, 3) and sends them to the interface. Then the reader extracts the lowermost six bits of the first column (4, 5, 6, 7, 8, 9) and sends them to the interface. This action is repeated for columns 2, 3, etc., up to column 80. No decoding is necessary at the card reader to convert the code from 12 bits to 6, as was the case for the alphanumeric system. The binary code allows twice as many (160) characters on a card as the alphanumeric system.

## Sensing and Control

Control and sense lines provided for operation of the card reader are listed in Table 3-5. As shown in the table, two timing modes are available: a continuous mode and a pulsed mode. In the continuous mode, the OCP signal will cause the cards to be read until any one of three conditions takes place: (1) the card hopper is empty, (2) a fault occurs in the card reader, or (3) a stop code on a card is read. In the pulsed mode a read-one-card OCP must be received each time a read operation is to take place.

Table 3-5.
Card Reader OCP and Sense Lines

| Line | Code | Description |
|---|---|---|
| OCP | '02400 | Read one Hollerith card |
| | '02401 | Read one binary card |
| | '02410 | Read Hollerith card continuously |
| | '02411 | Read binary cards continuously |
| | '02420 | Stop card reader |
| SENSE | '22400 | Card reader busy |
| | '22401 | End of file |
| | '22402 | Card reader not ready |

## 200 CPM Card-Reader Operation

Cards are placed face down in the feed hopper, with column one the first column to be read. If after the POWER ON pushbutton is pressed, the NOT READY light does not go off, check possible causes for this condition (see Table 3-6). If a check of the code validity is desired, the VALIDITY ON pushbutton should be pressed. Refer to Table 3-7 for a description of the controls.

Table 3-6.
200 CPM Card Reader Controls and Indicators

| Control/Indicator | Function |
|---|---|
| POWER ON | Applies primary power to system and indicates same |
| NOT READY | Indicates that one of the following conditions is present:<br>(a) Hopper empty<br>(b) Stacker full<br>(c) READ CHECK on<br>(d) VALIDITY CHECK on<br>(e) FEED CHECK on<br>(f) START switch has not been pressed<br>(g) STOP switch has been pressed |
| FEED CHECK | Indicates a card jam or feed problem |
| VALIDITY CHECK | An invalid character sensed during a VALIDITY ON condition |
| END OF FILE | Sets the end-of-file flip-flop; effective only when card hopper is empty |
| START | Places reader in operating mode ready for computer read instruction |
| POWER OFF | Removes power from reader |
| READ CHECK | Indicates that a read fault has taken place |
| VALIDITY ON | Causes reader to check for invalid characters and indicates that reader is in this check mode |
| RESET | Causes VALIDITY CHECK, READ CHECK, and FEED CHECK indicators to be reset, and, restarts the reader following a feed check. |
| STOP | Causes reader to go into NOT READY condition |

Table 3-7.
100 CPM Card Reader Controls and Indicators

| Control/Indicator | Function |
|---|---|
| READY | Indicates AC power on |
| READ CARD | Indicates card being read |
| ON/OFF | Power switch |
| MAN/AUTO | In AUTO position, enables automatic feed. In MAN position, disables automatic feed to permit entry of a single card |
| REG | Positions initial card for reading in AUTO (automatic) mode, and each card for reading in MAN (manual) mode |
| EOF | Sets end-of-file flip-flop, which may be queried by SKS '22401 |

## 100 CPM Card-Reader Operation

Cards are placed face down in the feed hopper, with row nine toward the rear of the reader, and the ON/OFF switch is pressed. The REG pushbutton is then pressed and held until card motion stops, and the reader is ready for operation. Table 3-7 contains a list of 100 cpm card reader controls and indicators.

## CARD PUNCH

A 100 cps card punch (Figure 3-9) is available with the DDP-124. It can be connected to any of the standard or optional output channels.

The card-punching operation is divided into four cycles - feeding, punching, checking, and stacking.

Cycle I - Feeding. After the POWER ON and START switches have been depressed, a card will feed into the ready station and be held there by the die card level.

Cycle II - Punching. Upon receipt of a feed and punch command, a card movement mechanism passes the card (12th row first, face down) under the line of 80 punches, stopping the card momentarily under each of the 12 rows. At the proper column positions, as determined by the punch code received from the computer, the card is punched. When a card in the ready position starts its movement for punching, a second card is fed into the ready station.

Cycle III - Checking. During the third cycle the card is fed through the post punch check brushes and checked for errors.

Cycle IV - Stacking. The card is placed on the output stack.

### Specifications

The specifications of the card punch are as follows.

Speed:

100 cards per minute

Feed:

Row-by-row punching
Corner turning from column by column to row by row done by program

### Sensing and Control

After the START button has been depressed the punch is under control of the computer program. Then, an OCP card punch select ('02500) can enable the card punch.

Computer punch character code transfer is accomplished in serial fashion, one row at a time. For example, the coding of the 80 digits in row 12 is done by shifting the code n, column by column, until the last column contains either a punch or a no punch signal. This requires 80 shifts prior to the punch strobe for that row. This process is repeated each time a new row is positioned under the punches until all 12 rows have been punched.

The computer uses three sense lines from the punch to monitor its operation as follows.

SKS '22500 - Punch busy. Present from the time a card is clutched until approximately 4 ms after the last row has been punched.

SKS '22501 - Ready to punch next row. Present while punching each row (approximately 820 μsec each time).

SKS '22502 - Punch not ready. Present while any of the not ready conditions in Table 3-8 are present.

Figure 3-9. 100 CPM Card Punch

Table 3-8.
Card Punch Controls

| Control | Function |
|---|---|
| POWER ON | Applies primary power to punch and indicates same |
| NOT READY | Indicates that one of the following conditions is present.<br>(a) Hopper empty<br>(b) Stacker full<br>(c) Clip box not in place<br>(d) A cover not in place<br>(e) Punch die improperly installed<br>(f) START switch has not been pressed<br>(g) STOP switch has been pressed |
| PUNCH CHECK | Indicates that code sent was not punched when in the PUNCH CHECK ON mode |
| FEED CHECK | Indicates a card jam or improper feed |
| START | Places punch in operating mode ready for computer punch instructions, and indicates same |
| POWER OFF | Removes primary power |
| RUNOUT | Causes punch to clear any cards in any of the four operating cycles when in NOT READY condition |
| PUNCH CHECK ON | Causes punch to check that codes received are codes punched |
| RESET | Resets error circuits |
| STOP | Puts punch in NOT READY mode without removing power |

Operating Notes

Cards are fed to the card punch face down with the 12th row first. (Refer to Table 3-8 for control functions.)

Since the card punch punches the cards row by row and the data on the cards has meaning only when considered by column, the entire card must be considered as a single block of output data, and a DDP-124 character card image must be converted to the bit configuration of an actual Hollerith or binary card image before the data can be transferred to the card. Since the interface circuitry to perform this card image transformation is not normally supplied with the card punch, this transformation must be done by program. This transformation process has acquired the name of "Corner Turning."

When the punch is enabled (OCP '2500), the card is punched, The card punch buffer accepts 80 bits and punches a row, then 80 more, and punches until all 12 rows are punched. The punch buffer accepts 80 bits out of each 14 words sent to the punch. It selects the six least significant of the 24 bits of each word starting with the most significant bit. When the punch buffer counts the 80th bit, it punches the row.

When transferring from memory to card, the row 12, column 1 bit is the most significant bit of the least significant 6 bits of the first word, and so on until all 168 card bits have been transferred. The last bit, row 9, column 80, is the next most significant bit of the least significant 6 bits of the 168th word outputted.

The output can be either 168 words in succession or can occur in 14-word blocks. In this case, an SKS '22501 (card punch ready to punch next row) must be coded between each block transferred.

## LINE PRINTER

A 300 lpm line printer (Figure 3-10).is available with the DDP-124. The line printer accepts coded signals from the computer and produces printed copy. The printer can approach a speed of 360 lines per minute printing alphanumeric characters only.

Printing is done by means of a rotating character drum and a bank of linear motion print hammers. The character drum contains 62 characters including the capital letters of the alphabet and all of the standard symbols commonly used on business and mathematical forms. Each symbol is repeated 120 times lengthwise across the cylindrical character drum. After the printer's buffer is full and the print OCP received by the printer, one full revolution of the character drum is needed to print the line. As each line of characters passes the print hammers the hammers corresponding to the data in the print buffer impact the paper against a printing ribbon and onto the appropriate character on the drum, thus printing all like characters at one time. When the print wheel revolution is complete, one full line has been printed.

Figure 3-10. Line Printer

## Information Transfer

The line printer is connected to the computer through a parallel output channel. (Refer to the DMA description in this section.) Transfer of data takes place with either OTM or OTA instruction, unless a DMA performs the transfer.

## Line Printer Operation

Line printer operation is a two-step process: loading the serial memory and printing. At other times, the printer is either waiting for data or advancing the paper. Upon receipt of a strobe from the timing and control circuits, the serial memory starts loading. If a print OCP arrives, the buffer will complete the loading of the last character sent to it, and the line will be printed with whatever characters have been loaded in printer storage. If a print OCP ('2302) does not arrive until after storage is full, the printer waits for the print OCP. If upon the receipt of a print OCP, the paper is still moving, printing is delayed until paper advance is complete. Consequently, in order to print, there must have been (1) data loaded into printer storage, (2) a print OCP signal, and (3) a PAPER ADVANCE COMPLETE signal.

## Format Control

The paper advance logic in the line printer buffer receives one of nine types of paper advance OCPs from the computer for each movement of the paper. (Refer to Table 3-9 for the OCP descriptions.) Paper advance in the line printer is controlled by a pre-programmed paper-tape loop on the paper-tape reader in the line printer.

Table 3-9.
Line Printer OCP Descriptions

| Code | Description |
|---|---|
| 02300 | Enable print buffer to accept 6-bit characters |
| 02301 | Enable print buffer to accept 24-bit words |
| 02302 | Print contents of buffer |
| 02310 | Advance carriage to punch in channel 1 of control tape |
| 02311 | Advance carriage to punch in channel 2 of control tape |
| 02312 | Advance carriage to punch in channel 3 of control tape |
| 02313 | Advance carriage to punch in channel 4 of control tape |
| 02314 | Advance carriage to punch in channel 5 of control tape |
| 02315 | Advance carriage to punch in channel 6 of control tape |
| 02316 | Advance carriage to punch in channel 7 of control tape |
| 02317 | Advance carriage to punch in channel 8 of control tape |
| 02304 | Advance carriage one line |

Track number 1 on the control tape is the top of form (OCP '02310) track and track number 8 is the single space (OCP '02317 or OCP '02304) track. These tracks are always used as the top of form and single space tracks; the other tracks may be punched as desired by the programmer to control the paper advance throughout the program. The paper advance logic receives the paper advance OCP signals from the computer and converts the OCP into a three-digit code that selects the desired track in the paper-tape loop. For each paper advance operation a PAPER ADVANCE signal is also fed to the printer, and upon completion of the operation an ADVANCE COMPLETE signal is returned to reset a control flip-flop in the advance logic.

### Character Codes

The line printer can print 61 different characters as determined by the 62 different character codes that are supplied to it by the computer. Acutally there are 64 character positions on the line printer drum, but two characters (the equal sign and apostrophe) are repeated on the drum and space code is not printed thereby resulting in a total of 61.

Table 3-1 indicates the DDP-124 code assigned to each printable character on the line printer.

## DIGITAL PLOTTER

A digital plotter may be connected to the computer through the 6-bit character buffer. The solid-state plotter records 100 incremental steps per inch, at the rate of 300 per second, along one or both of two perpendicular axes in either positive or negative directions. Pen-up and pen-down positions may change at a rate of 10 operations per second; limit switches control maximum excursion. Normal chart width is 12 in., but a 31-in. width is available. The plotter will plot points, continuous curves, curve identification symbols, letters, and numerals as directed by the DDP program.

## OPTION INTERFACE UNIT

An option interface unit (OIU) is available for housing logic for the optional data channels (DMA, WFB, etc.) that facilitate communication between the DDP-124 and peripheral devices. The OIU also includes OCP and SKS logic ordered by the user for controlling peripheral devices not supplied by 3C. (Peripheral equipment supplied by 3C always includes required OCP and SKS logic.)

# SECTION IV
# COMPUTER OPERATION

Each DDP-124 General-Purpose Computer is supplied with a control and maintenance panel to facilitate computer operation and maintenance. The panel is mounted on the optional console or on the main frame when no console is purchased. The control panel portion contains switches and indicators for use in controlling and monitoring computer operation. The maintenance panel portion contains controls and indicators for use in computer testing and trouble isolation and is not normally used for the control and evaluation of programs.

## CONTROL PANEL

The controls and indicators on the control panel are listed and described in Table 4-1. Controls and indicators have been divided into several groups for easy reference. The main registers of the computer are displayed on the control panel by means of pushbutton lights that show the binary contents of the registers and allow bit-by-bit entry. A red RESET button is provided adjacent to the indicator lights to clear each selected register to all ZEROs. To clear all display registers, depress the MASTER CLEAR pushbutton.

Table 4-1.
Control Panel Controls and Indicators

| Control Group | Control or Indicator | Function |
|---|---|---|
| Sense Switches | 1 through 6 | These switches provide manual control of programs. Control depends upon the instructions of the program. The state of any or all six switches can be tested through the use of properly coded SKS commands, after which branching may or may not take place, as determined by the setting of the switch. |
| Operational Indicators | RUN | Indicates that the halt flip-flop is reset and that operations can take place. |
| | INPUT/OUTPUT HOLD | Indicates that the computer is delaying until an I/O device becomes ready. |
| | INPUT PARITY | Allows detection of an error while reading character information; corresponds to state of input parity flip-flop. Flip-flop can be reset with properly coded SKS command. |
| | INTERRUPT | Indicates the execution of an interrupt subroutine by showing the state of the interrupt flip-flop. |

Table 4-1. (Cont)
Control Panel Controls and Indicators

| Control Group | Control or Indicator | Function |
|---|---|---|
| Operational Indicators (Cont) | OVERFLOW | Indicates that an overflow fault has occurred in the computation and that the overflow flip-flop is set. An overflow condition may occur in the following cases.<br>(a) When additions or subtractions produce sums or differences exceeding the capacity of the A-register. This will happen for additions or subtractions, and also for rounding of A with the RND command.<br>(b) During multiple precision operations when a carry is produced by the addition or subtraction of the next lower order portion. The overflow flip-flop will be reset after the complete multiple precision addition or subtraction has been performed correctly.<br>(c) When certain left shift instructions cause information to be shifted out of the A-register.<br>The OVERFLOW indicator may be reset by the JOF (jump on overflow) command or the SKS (skip if sense line not set) command. |
| | IMPROPER DIVIDE | Indicates the following:<br>(a) A division operation in which a numerator in the A-register is larger than or equal to the denominator (magnitude only). In this case, the resulting quotient surpasses the capacity of the B-register. The improper divide indicator may be reset by an SKS command. |
| Operational Controls | POWER | This pushbutton-indicator turns power on and off to the main computer and to auxiliary equipment. When power is switched on, the pushbutton is illuminated and the system normalizers reset the control flip-flops, preventing program runaway. No warmup period is required before the computer can be operated. |
| | HALT | Sets the halt flip-flop to stop the computer after the current instruction is completed. |
| | START | Causes the clock to run. If the FILL/HLT/RUN selector is in the run position the computer will start, beginning with the command stored in the memory location indicated by the program register. If this selector is in the HLT position the start button will cause the computer to do a single operation. If this selector is in the FILL position the start button will cause the computer to fill information from a paper tape into consecutive locations in memory. |
| | EXECUTE | Used in conjunction with the start button to cause the computer to execute the command set in the O-register. The operand address, if required, must be placed in the address portion of the Z-register, bits 10 through 24. The program register is not incremented; however, executing a jump or skip instruction will alter the program register. |

Table 4-1. (Cont)
Control Panel Controls and Indicators

| Control Group | Control or Indicator | Function |
|---|---|---|
| Operational Controls (Cont) | FETCH | Used in conjunction with the FILL/HLT/RUN selector and the start button. With the selector in the HLT position and depressing the start button causes the computer to fetch the contents of the memory location specified by the program register and places the information in the Z- and O-registers. The program register will be incremented by one during each fetch operation. If the selector is in the run position and the start button is depressed the computer will do continuous fetches from consecutive locations in memory. |
| | MASTER CLEAR | Sets the halt flip-flop; resets the following; the A- and B-registers and the index register, program register, control unit clock, ready and enable flip-flops of all input/output channels, the interrupt enable, input parity, overflow, and improper divide flip-flops. Also stops all peripheral devices. |
| | FILL | Selects the fill mode of operation. If the start button is depressed information from paper tape loads the computer memory in consecutive locations. |
| | FILL Selection switch | In the cont position, allows continuous operation of the paper tape reader. In the pulsed position, causes pulsed operation of the paper tape reader. |

Manual Program Loading

1. Put the EXECUTE/FETCH switch in the EXECUTE position.
2. Press the MASTER CLEAR button.
3. Set the A-register according to the word to be entered into the computer.
4. Set the OP-code register to the STA command (code 05), and the Z-register, bits 10 through 24, to the address of the word to be stored.
5. Put the HLT/RUN switch in the HLT position.
6. Press the start button.

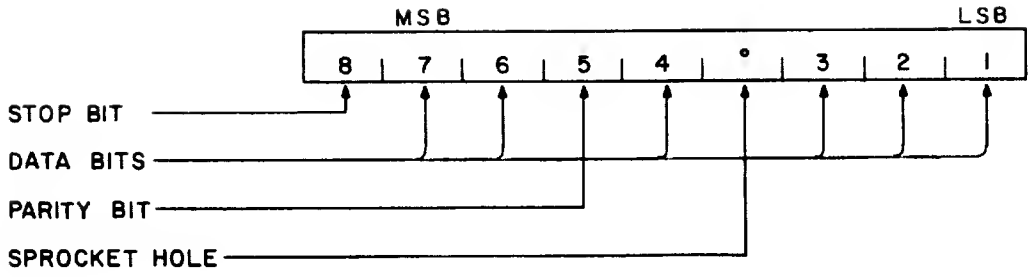
NOTE

This procedure loads only one word at a time; the effective address must be updated manually for consecutive words.

## PAPER-TAPE FORMAT

Paper-tape coding is indicated below. Punched holes in any of the eight paper-tape channels correspond to ONE bits. Paper-tape channel no. 1 corresponds to the least significant character bit.

When reading paper tape in octal format, only the bits in channels 1, 2, and 3 are read. During paper tape fill under control of the FILL button on the control panel, the bits in channels 4, 6, 7, and 8 must be ZERO. If any of these bits is ONE, the associated character is ignored.




Paper-Tape Format

When reading paper tape in binary or bi-octal-format, the bits in channels 1, 2, 3, 4, 6, and 7 are read. The parity bit is automatically checked while reading or generated while punching by the parity logic in the character buffer or word buffer. Odd parity is used.

If channel 8 contains a ONE bit (stop code) and channel 5 contains a ZERO (no punch), the paper-tape reader is stopped. The contents of channels 1 through 7 are read by the computer, and the stop code flip-flop is set. If both channels 8 and 5 contain a ONE bit, the character is ignored (channels 1 through 4, 6 and 7 are ignored), and the tape reader is not stopped. A stop code can be punched after setting the stop code flip-flop with an OCP command. The occurrence of a stop code from the paper-tape reader can be tested with an SKS command.

When reading or punching paper tape in the optionally available eight-channel mode (OCP 2140 and 2240, respectively), the parity and stop code positions are transferred as actual data.

# APPENDIX A
# OCP CODES

## I/O Buffer and Channel Lines

| | |
|---|---|
| 00000 | Enable Standard I/O Character Buffer |
| 00041 | Enable I/O Character Buffer No. 2 |
| 00042 | Enable I/O Character Buffer No. 3 |
| 00043 | Enable I/O Character Buffer No. 4 |
| 00044 | Enable I/O Character Buffer No. 5 |
| 00045 | Enable I/O Character Buffer No. 6 |
| 00046 | Enable I/O Character Buffer No. 7 |
| 00047 | Enable I/O Character Buffer No. 8 |
| 00001 | Enable Parallel Input Channel No. 1 |
| 00003 | Enable Parallel Input Channel No. 2 |
| 00005 | Enable Parallel Input Channel No. 3 |
| 00007 | Enable Parallel Input Channel No. 4 |
| 00011 | Enable Parallel Input Channel No. 5 |
| 00013 | Enable Parallel Input Channel No. 6 |
| 00015 | Enable Parallel Input Channel No. 7 |
| 00017 | Enable Parallel Input Channel No. 8 |
| 00002 | Enable Parallel Output Channel No. 1 |
| 00004 | Enable Parallel Output Channel No. 2 |
| 00006 | Enable Parallel Output Channel No. 3 |
| 00010 | Enable Parallel Output Channel No. 4 |
| 00012 | Enable Parallel Output Channel No. 5 |
| 00014 | Enable Parallel Output Channel No. 6 |
| 00016 | Enable Parallel Output Channel No. 7 |
| 00020 | Enable Parallel Output Channel No. 8 |
| 00021 | Enable Parallel I/O Channel No. 1 |
| 00022 | Enable Parallel I/O Channel No. 2 |
| 00023 | Enable Parallel I/O Channel No. 3 |
| 00024 | Enable Parallel I/O Channel No. 4 |
| 00025 | Enable Parallel I/O Channel No. 5 |
| 00026 | Enable Parallel I/O Channel No. 6 |
| 00027 | Enable Parallel I/O Channel No. 7 |
| 00030 | Enable Parallel I/O Channel No. 8 |
| 00031 | Enable Input Character Buffer No. 1 |
| 00033 | Enable Input Character Buffer No. 2 |
| 00035 | Enable Input Character Buffer No. 3 |
| 00037 | Enable Input Character Buffer No. 4 |
| 00032 | Enable Output Character Buffer No. 1 |
| 00034 | Enable Output Character Buffer No. 2 |
| 00036 | Enable Output Character Buffer No. 3 |
| 00040 | Enable Output Character Buffer No. 4 |
| 00077 | Disable All I/O Channels |

## WFB Lines

| | |
|---|---|
| 00320 | Eanble WFB No. 1 For 2 Char/Word Output |
| 00322 | Enable WFB No. 2 For 2 Char/Word Output |
| 00324 | Enable WFB No. 3 For 2 Char/Word Output |
| 00326 | Enable WFB No. 4 For 2 Char/Word Output |
| 00330 | Enable WFB No. 1 For 3 Char/Word Output |

## WFB Lines (Cont)

| | |
|---|---|
| 00332 | Enable WFB No. 2 For 3 Char/Word Output |
| 00334 | Enable WFB No. 3 For 3 Char/Word Output |
| 00336 | Enable WFB No. 4 For 3 Char/Word Output |
| 00340 | Enable WFB No. 1 For 4 Char/Word Output |
| 00342 | Enable WFB No. 2 For 4 Char/Word Output |
| 00344 | Enable WFB No. 3 For 4 Char/Word Output |
| 00346 | Enable WFB No. 4 For 4 Char/Word Output |
| 00321 | Enable WFB No. 1 For 2 Char/Word Input |
| 00323 | Enable WFB No. 2 For 2 Char/Word Input |
| 00325 | Enable WFB No. 3 For 2 Char/Word Input |
| 00327 | Enable WFB No. 4 For 2 Char/Word Input |
| 00331 | Enable WFB No. 1 For 3 Char/Word Input |
| 00333 | Enable WFB No. 2 For 3 Char/Word Input |
| 00335 | Enable WFB No. 3 For 3 Char/Word Input |
| 00337 | Enable WFB No. 4 For 3 Char/Word Input |
| 00341 | Enable WFB No. 1 For 4 Char/Word Input |
| 00343 | Enable WFB No. 2 For 4 Char/Word Input |
| 00345 | Enable WFB No. 3 For 4 Char/Word Input |
| 00347 | Enable WFB No. 4 For 4 Char/Word Input |
| 00310 | Enable WFB No. 1 For 1 Char/Word Output |
| 00312 | Enable WFB No. 2 For 1 Char/Word Output |
| 00314 | Enable WFB No. 3 For 1 Char/Word Output |
| 00316 | Enable WFB No. 4 For 1 Char/Word Output |
| 00311 | Enable WFB No. 1 For 1 Char/Word Input |
| 00313 | Enable WFB No. 2 For 1 Char/Word Input |
| 00315 | Enable WFB No. 3 For 1 Char/Word Input |
| 00317 | Enable WFB No. 4 For 1 Char/Word Input |
| 00412 | Enable WFB No. 2 For 1 Char/Word Output in 8-Bit Mode |
| 00422 | Enable WFB No. 2 For 2 Char/Word Output in 8-Bit Mode |
| 00432 | Enable WFB No. 2 For 3 Char/Word Output in 8-Bit Mode |
| 00413 | Enable WFB No. 2 For 1 Char/Word Input in 8-Bit Mode |
| 00423 | Enable WFB No. 2 For 2 Char/Word Input in 8-Bit Mode |
| 00433 | Enable WFB No. 2 For 3 Char/Word Input in 8-Bit Mode |
| 00512 | Enable WFB No. 2 For 1 Char/Word Output in 12-Bit Mode |
| 00522 | Enable WFB No. 2 For 2 Char/Word Output in 12-Bit Mode |
| 00513 | Enable WFB No. 2 For 1 Char/Word Input in 12-Bit Mode |
| 00523 | Enable WFB No. 2 For 2 Char/Word Input in 12-Bit Mode |

## Selective Interrupt Lines

| | |
|---|---|
| 02540 | Inhibit Interrupt On Line No. 0 |
| 02541 | Remove Inhibit From Line No. 0 |
| 02542 | Inhibit Interrupt On Line No. 1 |
| 02543 | Remove Inhibit From Line No. 1 |
| 02544 | Inhibit Interrupt On Line No. 2 |
| 02545 | Remove Inhibit From Line No. 2 |
| 02546 | Inhibit Interrupt On Line No. 3 |
| 02547 | Remove Inhibit From Line No. 3 |
| 02550 | Inhibit Interrupt On Line No. 4 |
| 02551 | Remove Inhibit From Line No. 4 |
| 02552 | Inhibit Interrupt On Line No. 5 |
| 02553 | Remove Inhibit From Line No. 5 |
| 02554 | Inhibit Interrupt On Line No. 6 |
| 02555 | Remove Inhibit From Line No. 6 |
| 02556 | Inhibit Interrupt On Line No. 7 |
| 02557 | Remove Inhibit From Line No. 7 |
| 02560 | Inhibit Interrupt On Line No. 8 |
| 02561 | Remove Inhibit From Line No. 8 |

## Selective Interrupt Lines (Cont)

| | |
|---|---|
| 02562 | Inhibit Interrupt On Line No. 9 |
| 02563 | Remove Inhibit From Line No. 9 |
| 02564 | Inhibit Interrupt On Line No. 10 |
| 02565 | Remove Inhibit From Line No. 10 |
| 02566 | Inhibit Interrupt On Line No. 11 |
| 02567 | Remove Inhibit From Line No. 11 |
| 02570 | Inhibit Interrupt On Line No. 12 |
| 02571 | Remove Inhibit From Line No. 12 |
| 02572 | Inhibit Interrupt On Line No. 13 |
| 02573 | Remove Inhibit From Line No. 13 |
| 02574 | Inhibit Interrupt On Line No. 14 |
| 02575 | Remove Inhibit From Line No. 14 |
| 02576 | Inhibit Interrupt On Line No. 15 |
| 02577 | Remove Inhibit From Line No. 15 |

## Optional OCP Lines

| | |
|---|---|
| 01010 | Optional OCP Line No. 1 |
| 01011 | Optional OCP Line No. 2 |
| 01012 | Optional OCP Line No. 3 |
| 01013 | Optional OCP Line No. 4 |
| 01014 | Optional OCP Line No. 5 |
| 01015 | Optional OCP Line No. 6 |
| 01016 | Optional OCP Line No. 7 |
| 01017 | Optional OCP Line No. 8 |
| 01020 | Optional OCP Line No. 9 |
| 01021 | Optional OCP Line No. 10 |
| 01022 | Optional OCP Line No. 11 |
| 01023 | Optional OCP Line No. 12 |
| 01024 | Optional OCP Line No. 13 |
| 01025 | Optional OCP Line No. 14 |
| 01026 | Optional OCP Line No. 15 |
| 01027 | Optional OCP Line No. 16 |
| 01030 | Optional OCP Line No. 17 |
| 01031 | Optional OCP Line No. 18 |
| 01032 | Optional OCP Line No. 19 |
| 01033 | Optional OCP Line No. 20 |
| 01034 | Optional OCP Line No. 21 |
| 01035 | Optional OCP Line No. 22 |
| 01036 | Optional OCP Line No. 23 |
| 01037 | Optional OCP Line No. 24 |
| 01040 | Optional OCP Line No. 25 |
| 01041 | Optional OCP Line No. 26 |
| 01042 | Optional OCP Line No. 27 |
| 01043 | Optional OCP Line No. 28 |
| 01044 | Optional OCP Line No. 29 |
| 01045 | Optional OCP Line No. 30 |
| 01046 | Optional OCP Line No. 31 |
| 01047 | Optional OCP Line No. 32 |

## Standard I/O Equipment Lines

| | |
|---|---|
| 01000 | Punch Stop Code On Paper-Tape Punch |
| 02000 | Typewriter Input Select And Character Buffer Enable |
| 02010 | Typewriter Output Select And Character Buffer Enable |
| 02070 | Disconnect All Three I/O Devices (Typewriter, Paper Tape Reader, And Paper Tape Punch) |
| 02100 | Enable Paper-Tape Reader And Character Buffer |

## Standard I/O Equipment Lines (Cont)

| | |
|---|---|
| 02200 | Enable Paper-Tape Punch And Character Buffer |
| 02210 | Turn Punch Power Off |
| 02220 | Feed One Sprocket Hold On Paper-Tape Punch |

## Line Printer Lines

| | |
|---|---|
| 00050 | Restore Line Printer Buffer Enable |
| 02300 | Enable Line Printer Buffer For Character Mode |
| 02301 | Enable Line Printer Buffer For Word Mode |
| 02302 | Print One Line On Line Printer |
| 02304 | Advance Paper One Line On Line Printer |
| 02310 | Advance Paper To Punch In Channel 1 On Line Printer Top Of Form |
| 02311 | Advance Paper To Punch In Channel 2 On Line Printer |
| 02312 | Advance Paper To Punch In Channel 3 On Line Printer |
| 02313 | Advance Paper To Punch In Channel 4 On Line Printer |
| 02314 | Advance Paper To Punch In Channel 5 On Line Printer |
| 02315 | Advance Paper To Punch In Channel 6 On Line Printer |
| 02316 | Advance Paper To Punch In Channel 7 On Line Printer |
| 02317 | Advance Paper To Punch In Channel 8 On Line Printer |

## Card Reader Lines

| | |
|---|---|
| 02400 | Read One Hollerith Card On Card Reader |
| 02401 | Read One Binary Card On Card Reader |
| 02410 | Read Hollerith Cards Continuously On Card Reader |
| 02411 | Read Binary Cards Continuously On Card Reader |
| 02420 | Stop Card Reader |

## Card Punch Lines

| | |
|---|---|
| 02500 | Select Card Punch |

## Digital Plotter Lines

| | |
|---|---|
| 02601 | Step -Y Or Carriage Right On Digital Plotter No. 1 |
| 02602 | Step +Y Or Carriage Left On Digital Plotter No. 1 |
| 02604 | Step -X Or Drum Up On Digital Plotter No. 1 |
| 02605 | Step -X And -Y On Digital Plotter No. 1 |
| 02606 | Step -X And +Y On Digital Plotter No. 1 |
| 02610 | Step +X Or Drum Down On Digital Plotter No. 1 |
| 02611 | Step +X And -Y On Digital Plotter No. 1 |
| 02612 | Step +X And +Y On Digital Plotter No. 1 |
| 02620 | Plotter Pen Down On Digital Plotter No. 1 |
| 02640 | Plotter Pen Up On Digital Plotter No. 1 |
| 02701 | Step -Y Or Carriage Right On Digital Plotter No. 2 |
| 02702 | Step +Y Or Carriage Left On Digital Plotter No. 2 |
| 02704 | Step -X Or Drum Up On Digital Plotter No. 2 |
| 02705 | Step -X And -Y On Digital Plotter No. 2 |
| 02706 | Step -X And +Y On Digital Plotter No. 2 |
| 02710 | Step +X Or Drum Down On Digital Plotter No. 2 |
| 02711 | Step +X And -Y On Digital Plotter No. 2 |
| 02712 | Step +X And +Y On Digital Plotter No. 2 |
| 02720 | Plotter Pen Down On Digital Plotter No. 2 |
| 02740 | Plotter Pen Up On Digital Plotter No. 2 |

## Real-Time Clock Lines

| | |
|---|---|
| 05040 | Start Real-Time Clock |
| 05041 | Stop Real-Time Clock |
| 05042 | Enable Output Channel To Real-Time Clock |
| 05043 | Enable Real-Time Clock To Input Channel |
| 05046 | Reset Real-Time Clock |
| 05047 | Disable Input And Output Channels To Real-Time Clock |
| 05050 | Connect Interval Timer Of Real-Time Clock To Interrupt |
| 05051 | Disconnect Interval Timer Of Real-Time Clock To Interrupt |
| 05052 | Connect Elapsed Timer Of Real-Time Clock To Interrupt |
| 05053 | Disconnect Elapsed Timer Of Real-Time Clock To Interrupt |

## DMA Lines

| | |
|---|---|
| 06000 | Enable DMA No. 1 |
| 06001 | Enable DMA No. 2 |
| 06002 | Enable DMA No. 3 |
| 06003 | Enable DMA No. 4 |

## Magnetic Tape Lines

| | |
|---|---|
| 20000 | Read Odd Parity On TCU No. 1, MTT No. 1 |
| 20001 | Read Odd Parity On TCU No. 1, MTT No. 2 |
| 20002 | Read Odd Parity On TCU No. 1, MTT No. 3 |
| 20003 | Read Odd Parity On TCU No. 1, MTT No. 4 |
| 20010 | Read Odd Parity On TCU No. 2, MTT No. 1 |
| 20011 | Read Odd Parity On TCU No. 2, MTT No. 2 |
| 20012 | Read Odd Parity On TCU No. 2, MTT No. 3 |
| 20013 | Read Odd Parity On TCU No. 2, MTT No. 4 |
| 20020 | Read Odd Parity On TCU No. 3, MTT No. 1 |
| 20021 | Read Odd Parity On TCU No. 3, MTT No. 2 |
| 20022 | Read Odd Parity On TCU No. 3, MTT No. 3 |
| 20023 | Read Odd Parity On TCU No. 3, MTT No. 4 |
| 20030 | Read Odd Parity On TCU No. 4, MTT No. 1 |
| 20031 | Read Odd Parity On TCU No. 4, MTT No. 2 |
| 20032 | Read Odd Parity On TCU No. 4, MTT No. 3 |
| 20033 | Read Odd Parity On TCU No. 4, MTT No. 4 |
| 20100 | Read Even Parity On TCU No. 1, MTT No. 1 |
| 20101 | Read Even Parity On TCU No. 1, MTT No. 2 |
| 20102 | Read Even Parity On TCU No. 1, MTT No. 3 |
| 20103 | Read Even Parity On TCU No. 1, MTT No. 4 |
| 20110 | Read Even Parity On TCU No. 2, MTT No. 1 |
| 20111 | Read Even Parity On TCU No. 2, MTT No. 2 |
| 20112 | Read Even Parity On TCU No. 2, MTT No. 3 |
| 20113 | Read Even Parity On TCU No. 2, MTT No. 4 |
| 20120 | Read Even Parity On TCU No. 3, MTT No. 1 |
| 20121 | Read Even Parity On TCU No. 3, MTT No. 2 |
| 20122 | Read Even Parity On TCU No. 3, MTT No. 3 |
| 20123 | Read Even Parity On TCU No. 3, MTT No. 4 |
| 20130 | Read Even Parity On TCU No. 4, MTT No. 1 |
| 20131 | Read Even Parity On TCU No. 4, MTT No. 2 |
| 20132 | Read Even Parity On TCU No. 4, MTT No. 3 |
| 20133 | Read Even Parity On TCU No. 4, MTT No. 4 |
| 20200 | Write Odd Parity On TCU No. 1, MTT No. 1 |
| 20201 | Write Odd Parity On TCU No. 1, MTT No. 2 |
| 20202 | Write Odd Parity On TCU No. 1, MTT No. 3 |
| 20203 | Write Odd Parity On TCU No. 1, MTT No. 4 |

Magnetic Tape Lines (Cont)

| | |
|---|---|
| 20210 | Write Odd Parity On TCU No. 2, MTT No. 1 |
| 20211 | Write Odd Parity On TCU No. 2, MTT No. 2 |
| 20212 | Write Odd Parity On TCU No. 2, MTT No. 3 |
| 20213 | Write Odd Parity On TCU No. 2, MTT No. 4 |
| 20220 | Write Odd Parity On TCU No. 3, MTT No. 1 |
| 20221 | Write Odd Parity On TCU No. 3, MTT No. 2 |
| 20222 | Write Odd Parity On TCU No. 3, MTT No. 3 |
| 20223 | Write Odd Parity On TCU No. 3, MTT No. 4 |
| 20230 | Write Odd Parity On TCU No. 4, MTT No. 1 |
| 20231 | Write Odd Parity On TCU No. 4, MTT No. 2 |
| 20232 | Write Odd Parity On TCU No. 4, MTT No. 3 |
| 20233 | Write Odd Parity On TCU No. 4, MTT No. 4 |
| 20300 | Write Even Parity On TCU No. 1, MTT No. 1 |
| 20301 | Write Even Parity On TCU No. 1, MTT No. 2 |
| 20302 | Write Even Parity On TCU No. 1, MTT No. 3 |
| 20303 | Write Even Parity On TCU No. 1, MTT No. 4 |
| 20310 | Write Even Parity On TCU No. 2, MTT No. 1 |
| 20311 | Write Even Parity On TCU No. 2, MTT No. 2 |
| 20312 | Write Even Parity On TCU No. 2, MTT No. 3 |
| 20313 | Write Even Parity On TCU No. 2, MTT No. 4 |
| 20320 | Write Even Parity On TCU No. 3, MTT No. 1 |
| 20321 | Write Even Parity On TCU No. 3, MTT No. 2 |
| 20322 | Write Even Parity On TCU No. 3, MTT No. 3 |
| 20323 | Write Even Parity On TCU No. 3, MTT No. 4 |
| 20330 | Write Even Parity On TCU No. 4, MTT No. 1 |
| 20331 | Write Even Parity On TCU No. 4, MTT No. 2 |
| 20332 | Write Even Parity On TCU No. 4, MTT No. 3 |
| 20333 | Write Even Parity On TCU No. 4, MTT No. 4 |
| 20400 | Write End Of File On TCU No. 1, MTT No. 1 |
| 20401 | Write End Of File On TCU No. 1, MTT No. 2 |
| 20402 | Write End Of File On TCU No. 1, MTT No. 3 |
| 20403 | Write End Of File On TCU No. 1, MTT No. 4 |
| 20410 | Write End Of File On TCU No. 2, MTT No. 1 |
| 20411 | Write End Of File On TCU No. 2, MTT No. 2 |
| 20412 | Write End Of File On TCU No. 2, MTT No. 3 |
| 20413 | Write End Of File On TCU No. 2, MTT No. 4 |
| 20420 | Write End Of File On TCU No. 3, MTT No. 1 |
| 20421 | Write End Of File On TCU No. 3, MTT No. 2 |
| 20422 | Write End Of File On TCU No. 3, MTT No. 3 |
| 20423 | Write End Of File On TCU No. 3, MTT No. 4 |
| 20430 | Write End Of File On TCU No. 4, MTT No. 1 |
| 20431 | Write End Of File On TCU No. 4, MTT No. 2 |
| 20432 | Write End Of File On TCU No. 4, MTT No. 3 |
| 20433 | Write End Of File On TCU No. 4, MTT No. 4 |
| 30000 | Backspace One Record On TCU No. 1, MTT No. 1 |
| 30001 | Backspace One Record On TCU No. 1, MTT No. 2 |
| 30002 | Backspace One Record On TCU No. 1, MTT No. 3 |
| 30003 | Backspace One Record On TCU No. 1, MTT No. 4 |
| 30010 | Backspace One Record On TCU No. 2, MTT No. 1 |
| 30011 | Backspace One Record On TCU No. 2, MTT No. 2 |
| 30012 | Backspace One Record On TCU No. 2, MTT No. 3 |
| 30013 | Backspace One Record On TCU No. 2, MTT No. 4 |
| 30020 | Backspace One Record On TCU No. 3, MTT No. 1 |
| 30021 | Backspace One Record On TCU No. 3, MTT No. 2 |
| 30022 | Backspace One Record On TCU No. 3, MTT No. 3 |
| 30023 | Backspace One Record On TCU No. 3, MTT No. 4 |
| 30030 | Backspace One Record On TCU No. 4, MTT No. 1 |
| 30031 | Backspace One Record On TCU No. 4, MTT No. 2 |
| 30032 | Backspace One Record On TCU No. 4, MTT No. 3 |
| 30033 | Backspace One Record On TCU No. 4, MTT No. 4 |

## Magnetic Tape Lines (Cont)

| | |
|---|---|
| 30100 | Backspace One File On TCU No. 1, MTT No. 1 |
| 30101 | Backspace One File On TCU No. 1, MTT No. 2 |
| 30102 | Backspace One File On TCU No. 1, MTT No. 3 |
| 30103 | Backspace One File On TCU No. 1, MTT No. 4 |
| 30110 | Backspace One File On TCU No. 2, MTT No. 1 |
| 30111 | Backspace One File On TCU No. 2, MTT No. 2 |
| 30112 | Backspace One File On TCU No. 2, MTT No. 3 |
| 30113 | Backspace One File On TCU No. 2, MTT No. 4 |
| 30120 | Backspace One File On TCU No. 3, MTT No. 1 |
| 30121 | Backspace One File On TCU No. 3, MTT No. 2 |
| 30122 | Backspace One File On TCU No. 3, MTT No. 3 |
| 30123 | Backspace One File On TCU No. 3, MTT No. 4 |
| 30130 | Backspace One File On TCU No. 4, MTT No. 1 |
| 30131 | Backspace One File On TCU No. 4, MTT No. 2 |
| 30132 | Backspace One File On TCU No. 4, MTT No. 3 |
| 30133 | Backspace One File On TCU No. 4, MTT No. 4 |
| 30200 | Skip One Record On TCU No. 1, MTT No. 1 |
| 30201 | Skip One Record On TCU No. 1, MTT No. 2 |
| 30202 | Skip One Record On TCU No. 1, MTT No. 3 |
| 30203 | Skip One Record On TCU No. 1, MTT No. 4 |
| 30210 | Skip One Record On TCU No. 2, MTT No. 1 |
| 30211 | Skip One Record On TCU No. 2, MTT No. 2 |
| 30212 | Skip One Record On TCU No. 2, MTT No. 3 |
| 30213 | Skip One Record On TCU No. 2, MTT No. 4 |
| 30220 | Skip One Record On TCU No. 3, MTT No. 1 |
| 30221 | Skip One Record On TCU No. 3, MTT No. 2 |
| 30222 | Skip One Record On TCU No. 3, MTT No. 3 |
| 30223 | Skip One Record On TCU No. 3, MTT No. 4 |
| 30230 | Skip One Record On TCU No. 4, MTT No. 1 |
| 30231 | Skip One Record On TCU No. 4, MTT No. 2 |
| 30232 | Skip One Record On TCU No. 4, MTT No. 3 |
| 30233 | Skip One Record On TCU No. 4, MTT No. 4 |
| 30300 | Skip One File On TCU No. 1, MTT No. 1 |
| 30301 | Skip One File On TCU No. 1, MTT No. 2 |
| 30302 | Skip One File On TCU No. 1, MTT No. 3 |
| 30303 | Skip One File On TCU No. 1, MTT No. 4 |
| 30310 | Skip One File On TCU No. 2, MTT No. 1 |
| 30311 | Skip One File On TCU No. 2, MTT No. 2 |
| 30312 | Skip One File On TCU No. 2, MTT No. 3 |
| 30313 | Skip One File On TCU No. 2, MTT No. 4 |
| 30320 | Skip One File On TCU No. 3, MTT No. 1 |
| 30321 | Skip One File On TCU No. 3, MTT No. 2 |
| 30322 | Skip One File On TCU No. 3, MTT No. 3 |
| 30323 | Skip One File On TCU No. 3, MTT No. 4 |
| 30330 | Skip One File On TCU No. 4, MTT No. 1 |
| 30331 | Skip One File On TCU No. 4, MTT No. 2 |
| 30332 | Skip One File On TCU No. 4, MTT No. 3 |
| 30333 | Skip One File On TCU No. 4, MTT No. 4 |
| 30400 | Rewind On TCU No. 1, MTT No. 1 |
| 30401 | Rewind On TCU No. 1, MTT No. 2 |
| 30402 | Rewind On TCU No. 1, MTT No. 3 |
| 30403 | Rewind On TCU No. 1, MTT No. 4 |
| 30410 | Rewind On TCU No. 2, MTT No. 1 |
| 30411 | Rewind On TCU No. 2, MTT No. 2 |
| 30412 | Rewind On TCU No. 2, MTT No. 3 |
| 30413 | Rewind On TCU No. 2, MTT No. 4 |
| 30420 | Rewind On TCU No. 3, MTT No. 1 |
| 30421 | Rewind On TCU No. 3, MTT No. 2 |
| 30433 | Rewind On TCU No. 3, MTT No. 3 |
| 30423 | Rewind On TCU No. 3, MTT No. 4 |

Magnetic Tape Lines (Cont)

| | |
|---|---|
| 30430 | Rewind On TCU No. 4, MTT No. 1 |
| 30431 | Rewind On TCU No. 4, MTT No. 2 |
| 30432 | Rewind On TCU No. 4, MTT No. 3 |
| 30433 | Rewind On TCU No. 4, MTT No. 4 |

# APPENDIX B
# SKS CODES

## Simultaneous Test Lines

| | |
|---|---|
| 00001 | Sense Switch No. 1 |
| 00002 | Sense Switch No. 2 |
| 00004 | Sense Switch No. 3 |
| 00010 | Sense Switch No. 4 |
| 00020 | Sense Switch No. 5 |
| 00040 | Sense Switch No. 6 |
| 00100 | Parity Error From Standard I/O Character Buffer |
| 00200 | Improper Divide |
| 00400 | Overflow |
| 01000 | Stop Code |
| 02000 | Character Device (Typewriter, Paper Tape Reader, Paper Tape Punch) Busy |
| 14000 | Standard I/O Character Buffer Ready |

## Optional SKS Lines

| | |
|---|---|
| 20000 | Optional SKS Line No. 1 |
| 20001 | Optional SKS Line No. 2 |
| 20002 | Optional SKS Line No. 3 |
| 20003 | Optional SKS Line No. 4 |
| 20004 | Optional SKS Line No. 5 |
| 20005 | Optional SKS Line No. 6 |
| 20006 | Optional SKS Line No. 7 |
| 20007 | Optional SKS Line No. 8 |
| 20010 | Optional SKS Line No. 9 |
| 20011 | Optional SKS Line No. 10 |
| 20012 | Optional SKS Line No. 11 |
| 20013 | Optional SKS Line No. 12 |
| 20014 | Optional SKS Line No. 13 |
| 20015 | Optional SKS Line No. 14 |
| 20016 | Optional SKS Line No. 15 |
| 20017 | Optional SKS Line No. 16 |
| 20020 | Optional SKS Line No. 17 |
| 20021 | Optional SKS Line No. 18 |
| 20022 | Optional SKS Line No. 19 |
| 20023 | Optional SKS Line No. 20 |
| 20024 | Optional SKS Line No. 21 |
| 20025 | Optional SKS Line No. 22 |
| 20026 | Optional SKS Line No. 23 |
| 20027 | Optional SKS Line No. 24 |

## Line Printer Lines

| | |
|---|---|
| 22300 | Line Printer Busy |
| 22301 | Line Printer Parity Error |
| 22302 | Line Printer Paper Advancing |

### Card Reader Lines

| | |
|---|---|
| 22400 | Card Reader Busy |
| 22401 | Card Reader End of File |
| 22402 | Card Reader Not Ready |

### Card Punch Lines

| | |
|---|---|
| 22500 | Card Punch Busy |
| 22501 | Card Punch Ready to Punch Next Row |
| 22502 | Card Punch Not Ready |

### Real-Time Clock Lines

| | |
|---|---|
| 25042 | Real-Time Clock Busy |

### DMA Lines

| | |
|---|---|
| 26000 | DMA No. 1 Busy |
| 26001 | DMA No. 2 Busy |
| 26002 | DMA No. 3 Busy |
| 26003 | DMA No. 4 Busy |

### Magnetic Tape Lines

| | |
|---|---|
| 30300 | Parity Error On WFB No. 1 |
| 30302 | Parity Error On WFB No. 2 |
| 30304 | Parity Error On WFB No. 3 |
| 30306 | Parity Error On WFB No. 4 |
| 31000 | TCU No. 1 Busy |
| 31010 | TCU No. 2 Busy |
| 31020 | TCU No. 3 Busy |
| 31030 | TCU No. 4 Busy |
| 31100 | TCU No. 1, MTT No. 1 Rewinding |
| 31101 | TCU No. 1, MTT No. 2 Rewinding |
| 31102 | TCU No. 1, MTT No. 3 Rewinding |
| 31103 | TCU No. 1, MTT No. 4 Rewinding |
| 31110 | TCU No. 2, MTT No. 1 Rewinding |
| 31111 | TCU No. 2, MTT No. 2 Rewinding |
| 31112 | TCU No. 2, MTT No. 3 Rewinding |
| 31113 | TCU No. 2, MTT No. 4 Rewinding |
| 31120 | TCU No. 3, MTT No. 1 Rewinding |
| 31121 | TCU No. 3, MTT No. 2 Rewinding |
| 31122 | TCU No. 3, MTT No. 3 Rewinding |
| 31123 | TCU No. 3, MTT No. 4 Rewinding |
| 31130 | TCU No. 4, MTT No. 1 Rewinding |
| 31131 | TCU No. 4, MTT No. 2 Rewinding |
| 31132 | TCU No. 4, MTT No. 3 Rewinding |
| 31133 | TCU No. 4, MTT No. 4 Rewinding |
| 31200 | End of Tape On No. 1 TCU |
| 31210 | End of Tape On No. 2 TCU |
| 31220 | End of Tape On No. 3 TCU |
| 31230 | End of Tape On No. 4 TCU |
| 31300 | Beginning of Tape On No. 1 TCU |
| 31310 | Beginning of Tape On No. 2 TCU |
| 31320 | Beginning of Tape On No. 3 TCU |
| 31330 | Beginning of Tape On No. 4 TCU |

## Magnetic Tape Lines (Cont)

| | |
|---|---|
| 31400 | End of File On No. 1 TCU |
| 31410 | End of File On No. 2 TCU |
| 31420 | End of File On No. 3 TCU |
| 31430 | End of File On No. 4 TCU |

## I/O Channel Lines

| | |
|---|---|
| 37003 | Parallel Input Channel No. 2 Ready |
| 37005 | Parallel Input Channel No. 3 Ready |
| 37007 | Parallel Input Channel No. 4 Ready |
| 37011 | Parallel Input Channel No. 5 Ready |
| 37013 | Parallel Input Channel No. 6 Ready |
| 37015 | Parallel Input Channel No. 7 Ready |
| 37017 | Parallel Input Channel No. 8 Ready |
| 37004 | Parallel Output Channel No. 2 Ready |
| 37006 | Parallel Output Channel No. 3 Ready |
| 37010 | Parallel Output Channel No. 4 Ready |
| 37012 | Parallel Output Channel No. 5 Ready |
| 37014 | Parallel Output Channel No. 6 Ready |
| 37016 | Parallel Output Channel No. 7 Ready |
| 37020 | Parallel Output Channel No. 8 Ready |
| 37021 | Parallel I/O Channel No. 1 Ready |
| 37022 | Parallel I/O Channel No. 2 Ready |
| 37023 | Parallel I/O Channel No. 3 Ready |
| 37024 | Parallel I/O Channel No. 4 Ready |
| 37025 | Parallel I/O Channel No. 5 Ready |
| 37026 | Parallel I/O Channel No. 6 Ready |
| 37027 | Parallel I/O Channel No. 7 Ready |
| 37030 | Parallel I/O Channel No. 8 Ready |

## I/O Buffer Lines

| | |
|---|---|
| 37031 | Input Character Buffer No. 1 Ready |
| 37033 | Input Character Buffer No. 2 Ready |
| 37035 | Input Character Buffer No. 3 Ready |
| 37037 | Input Character Buffer No. 4 Ready |
| 37032 | Output Character Buffer No. 1 Ready |
| 37034 | Output Character Buffer No. 2 Ready |
| 37036 | Output Character Buffer No. 3 Ready |
| 37040 | Output Character Buffer No. 4 Ready |
| 37041 | I/O Character Buffer No. 2 Ready |
| 37042 | I/O Character Buffer No. 3 Ready |
| 37043 | I/O Character Buffer No. 4 Ready |
| 37044 | I/O Character Buffer No. 5 Ready |
| 37045 | I/O Character Buffer No. 6 Ready |
| 37046 | I/O Character Buffer No. 7 Ready |
| 37047 | I/O Character Buffer No. 8 Ready |

## Parity Error Lines

| | |
|---|---|
| 37131 | Parity Error From Input Character Buffer No. 1 |
| 37133 | Parity Error From Input Character Buffer No. 2 |
| 37135 | Parity Error From Input Character Buffer No. 3 |
| 37137 | Parity Error From Input Character Buffer No. 4 |
| 37132 | Parity Error From Output Character Buffer No. 1 |
| 37134 | Parity Error From Output Character Buffer No. 2 |

## Parity Error Lines (Cont)

| | |
|---|---|
| 37136 | Parity Error From Output Character Buffer No. 3 |
| 37140 | Parity Error From Output Character Buffer No. 4 |
| 37141 | Parity Error From I/O Character Buffer No. 2 |
| 37142 | Parity Error From I/O Character Buffer No. 3 |
| 37143 | Parity Error From I/O Character Buffer No. 4 |
| 37144 | Parity Error From I/O Character Buffer No. 5 |
| 37145 | Parity Error From I/O Character Buffer No. 6 |
| 37146 | Parity Error From I/O Character Buffer No. 7 |
| 37147 | Parity Error From I/O Character Buffer No. 8 |

## Reset Lines

| | |
|---|---|
| 65040 | Reset Interrupt No. 1 On Real-Time Clock |
| 65041 | Reset Interrupt No. 2 On Real-Time Clock |

# APPENDIX C
# NUMERICAL COMMAND LIST

| OP Code | Mnemonic | Description | Execution Time |
|---|---|---|---|
| 00 | HLT | Halt | 1.75 μsec |
| 02 | XEC | Execute | 1.75 μsec+ |
| 03 | STB | Store B | 3.5 μsec |
| 05 | STA | Store A | 3.5 μsec |
| 06 | STD | Store Address Portion of A | 3.5 μsec |
| 07 | INM | Input to Memory | 3.5 μsec |
| 10 | ADD | Add | 3.5 μsec |
| 11 | SUB | Subtract | 3.5 μsec |
| 12 | SKG | Skip if A Greater | 4.6 μsec |
| 13 | SKN | Skip if A Not Equal | 4.6 μsec |
| 15 | ANA | AND to A | 3.5 μsec |
| 16 | ORA | OR to A | 3.5 μsec |
| 17 | ERA | Exclusive OR to A | 3.5 μsec |
| 22 | OTM | Output from Memory | 3.5 μsec |
| 23 | LDB | Load B | 3.5 μsec |
| 24 | LDA | Load A | 3.5 μsec |
| 25 | JRT | Jump Return | 3.5 μsec |
| 27 | JST | Jump and Store Location | 3.5 μsec |
| 30 | SMP | Step Multiple Precision | 3.5 μsec |
| 34 | MPY | Multiply | 14 μsec (avg) |
| 35 | DIV | Divide | 22 μsec |
| 40 | ARS | A Right Shift | 2.8 + x μsec |
| 41 | ALS | A Left Shift | 2.8 + x μsec |
| 42 | LRR | Long Right Rotate | 2.8 + x μsec |
| 43 | LLR | Long Left Rotate | 2.8 + x μsec |
| 44 | LRS | Long Right Shift | 2.8 + x μsec |
| 45 | LLS | Long Left Shift | 2.8 + x μsec |
| 47 | LGL | Logical Left Shift | 2.8 + x μsec |
| 50 | OTA | Output from A | 2.1 μsec (min) |
| 52 | INA | Input to A | 1.75 μsec |
| 53 | OCP | Output Control Pulse | 2.1 μsec |
| 54 | ADX | Add to Index | 1.75 μsec |
| 55 | TAB | Transfer A to B | 1.75 μsec |
| 56 | LDX | Load Index | 1.75 μsec |
| 57 | IAB | Interchange A and B | 1.75 μsec |
| 60 | CRA | Clear A | 1.75 μsec |
| 61 | SKS | Skip if Sense Line Not Set | 4.2 μsec |
| 62 | RND | Round A | 1.75 μsec |
| 63 | TAX | Transfer A to Index | 1.75 μsec |
| 66 | STX | Store Index | 3.5 μsec |
| 67 | IRX | Increment, Replace and Load Index | 5.25 μsec |
| 70 | JPL | Jump if A Plus | 2.1 μsec |
| 71 | JZE | Jump if A Zero | 2.1 μsec |
| 72 | JIX | Jump on Index | 2.1 μsec |
| 73 | JOF | Jump on Overflow | 2.1 μsec |
| 74 | JMP | Unconditional Jump | 2.1 μsec |
| 75 | JXI | Jump on Index Incremented | 2.8 μsec |
| 77 | NOP | No Operation | 1.75 μsec |

# APPENDIX D
# ALPHABETICAL COMMAND LIST

| Mnemonic | Op Code | Description | Execution Time |
|---|---|---|---|
| ADD | 10 | Add | 3.5 μsec |
| ADX | 54 | Add to Index | 1.75 μsec |
| ALS | 41 | A Left Shift | 2.8 + x μsec |
| ANA | 15 | AND to A | 3.5 μsec |
| ARS | 40 | A Right Shift | 2.8 + x μsec |
| CRA | 60 | Clear A | 1.75 μsec |
| DIV | 35 | Divide | 22 μsec |
| ERA | 17 | Exclusive OR to A | 3.5 μsec |
| HLT | 00 | Halt | 1.75 μsec |
| IAB | 57 | Interchange A and B | 1.75 μsec |
| INA | 52 | Input to A | 1.75 μsec |
| INM | 07 | Input to Memory | 3.5 μsec |
| IRX | 67 | Increment, Replace and Load Index | 5.25 μsec |
| JIX | 72 | Jump on Index | 2.1 μsec |
| JMP | 74 | Unconditional Jump | 2.1 μsec |
| JOF | 73 | Jump on Overflow | 2.1 μsec |
| JPL | 70 | Jump if A Plus | 2.1 μsec |
| JRT | 25 | Jump Return | 3.5 μsec |
| JST | 27 | Jump and Store Location | 3.5 μsec |
| JXI | 75 | Jump on Index Incremented | 2.8 μsec |
| JZE | 71 | Jump if A Zero | 2.1 μsec |
| LDA | 24 | Load A | 3.5 μsec |
| LDB | 23 | Load B | 3.5 μsec |
| LDX | 56 | Load Index | 1.75 μsec |
| LGL | 47 | Logical Left Shift | 2.8 + x μsec |
| LLR | 43 | Long Left Rotate | 2.8 + x μsec |
| LLS | 45 | Long Left Shift | 2.8 + x μsec |
| LRR | 42 | Long Right Rotate | 2.8 + x μsec |
| LRS | 44 | Long Right Shift | 2.8 + x μsec |
| MPY | 34 | Multiply | 14 μsec (avg) |
| NOP | 77 | No Operation | 1.75 μsec |
| OCP | 53 | Output Control Pulse | 2.1 μsec |
| ORA | 16 | OR to A | 3.5 μsec |
| OTA | 50 | Output from A | 2.1 μsec (min) |
| OTM | 22 | Output from Memory | 3.5 μsec |
| RND | 62 | Round A | 1.75 μsec |
| SKG | 12 | Skip if A Greater | 4.6 μsec |
| SKN | 13 | Skip if A Not Equal | 4.6 μsec |
| SKS | 61 | Skip if Sense Line Not Set | 4.2 μsec |
| SMP | 30 | Step Multiple Precision | 3.5 μsec |
| STA | 05 | Store A | 3.5 μsec |
| STB | 03 | Store B | 3.5 μsec |
| STD | 06 | Store Address Portion of A | 3.5 μsec |
| STX | 66 | Store Index | 3.5 μsec |
| SUB | 11 | Subtract | 3.5 μsec |
| TAB | 55 | Transfer A to B | 1.75 μsec |
| TAX | 63 | Transfer A to Index | 1.75 μsec |
| XEC | 02 | Execute | 1.75 μsec+ |

COMPUTER CONTROL COMPANY, INC.
OLD CONNECTICUT PATH. FRAMINGHAM. MASS.